REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 15-04-2003	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From - To)* xx-08-2001 to 15-04-2003

4. TITLE AND SUBTITLE MAC-Layer Modeling and Analysis for Exploring Self-Regulation Enhancements in 802.11-Based Wireless Networks	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Haggard, Michael L; Author	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dept of Computer Science and Engineering University of South Carolina Columbia, SC 29208	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dept of Computer Science and Engineering University of South Carolina Columbia, SC 29208	**10. SPONSOR/MONITOR'S ACRONYM(S)** USC, Dept CSE **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

A Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This thesis presents a simulation model created to accurately explore the range of responses of a WLAN supporting a fighting force moving across the battlefield. The objective has been to define a simulation model, built on a accurate analysis model of the fighting force units, which takes into consideration formations and movement techniques of this force on the battlefield. By subjecting this model to different movement patterns, traffic profiles, and packet volume, assessment of QoS data patterns can be used to make dynamic changes to the management of the network to maximize QoS under such conditions. Creation of such a model is an essential first step in being able to identify possible data patterns that are recognizable, and thus “codifiable” into a set of heuristic enhancements to the 802.11 MAC protocol and architecture—with the goal being to endow the WLAN with self-regulating capabilities. The model developed in this thesis is envisioned to significantly aid in the design of effective ubiquitous, wearable computing and wireless communications devices.

15. SUBJECT TERMS

Network Simulation; MAC; Media Access Control Layer; Computer Network; Simulation; Network Self-Regulation; Network Modeling; Wireless Network;

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Michael L. Haggard
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	Unclassified Unlimited	124	**19b. TELEPHONE NUMBER** *(include area code)* 703-703-8506

MAC-LAYER MODELING AND ANALYSIS FOR EXPLORING SELF-REGULATION ENHANCEMENTS IN 802.11-BASED WIRELESS NETWORKS

by

Michael Luther Haggard

Bachelor of Science
United States Military Academy, 1993

Submitted in Partial Fulfillment of the

Requirements for the Degree of Master of Science in the

Department of Computer Science and Engineering

College of Engineering and Information Technology

University of South Carolina

2003

Department of Computer Science and
Engineering
Director of Thesis

Department of Computer Science and
Engineering
2nd Reader

Department of Computer Science and
Engineering
3rd Reader

Dean of the Graduate School

TABLE OF CONTENTS

CHAPTER 1

STATEMENT OF THE RESEARCH THESIS

1.1 The Military Initiative

On October 12, 1999, the Chief of Staff of the Army (CSA), General Eric Shinseki published *The Army Vision* [SHI99], which detailed how the Army would meet the Nation's need for a deployable, agile, versatile, lethal, survivable, and sustainable land force. The CSA saw a need for a new type of Army unit, which is a mixture of the deployability of light infantry formation with the lethality of armor formations. That force structure is now called the Objective Force. The Objective Force is a full spectrum unit with the capability to perform missions from a Major Theater War, Stability and Support Operations, counter-terrorism, and Homeland security [USA01].

The Objective Force is causing a shift in Army tactics. Traditionally, Army units would maneuver to meet the enemy and develop the situation while in contact. This approach translated into a focus on defeating the enemy force with overwhelming fires and protecting the friendly soldier with more inches of armor. The new Army tactic is to develop the situation while out of contact, then maneuver to defeat the enemy at a time and place of our choice. The focus has now changed to defeating the enemy by having a superior understanding of the battlefield picture, making timely and accurate decisions based on that understanding, and disseminating orders to subordinate units for execution.

The Army leadership states the concept as the force's ability to see first, understand first, act first, and finish decisively at the strategic, operational and tactical levels [USA01].

To have this ability, the Objective Force must have Decision Superiority and superior Knowledge Management over it's opponent. To have Decision Superiority, the battlefield commander must be able to make decisions faster and better than his opponent and communicate these decisions to his subordinates. Knowledge Management is the management of the variables that contribute to Decision Superiority in order to achieve an optimum degree of decision-making ability. The Army's Chief Information Officer and Deputy Chief of Staff for Intelligence describe the metrics of Knowledge Management as the availability, accuracy, relevance, and compatibility of the data; the timeliness, capacity, assuredness, and security of the transport layer; and the clarity and adaptability of the presentation [NOP00].

Decision Superiority is composed of several elements governing how data is handled, one of which is the communications infrastructure. The Army seeks to improve the infrastructure of the network through the development of self-healing systems, automatic fault detection and dynamic bandwidth adjustment to increase the seamless connectivity. The infrastructure improvement is part of a larger effort to transform the force from one where information was stovepiped in the organization that developed it, to one where information is shared via a common network and transformed into knowledge the commander can use to make timely and accurate decisions [NOP00].

Part of the technology developed for the Army's information infrastructure is the Land Warrior system. Designed for use by individual infantry soldiers, Land Warrior is a collection of systems, consisting of weapons, computer, communications, sensors, vision, and global positioning systems assembled for the purpose of providing increased lethality, command and control, mobility, survivability, and situational awareness for the individual infantry soldier [ORD01].

The communications system for Land Warrior is a wireless Local Area Network (WLAN) with the Mesh Radio providing the communications medium [MSH02]. Each soldier is a node of the network, which allows any soldier's communications system to route packets to any other soldier without the need to transmit to a central base station. As shown in Figure 1, soldiers are initially grouped into squad networks. Two or three squads and the platoon leadership combine to create the platoon network. Accordingly, two to three platoons and the company leadership combine to create the company network cloud. This process continues up to battalion level and above. Land Warrior is being developed for use in an infantry division [MSH02].

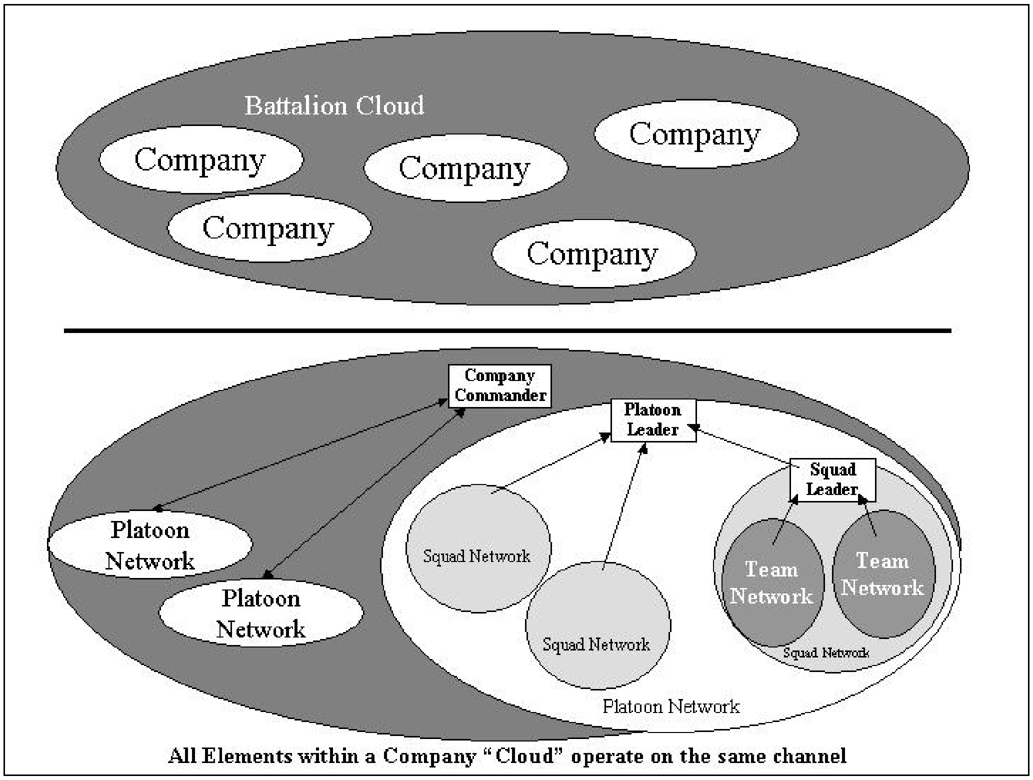


Figure 1: Topology of the Land Warrior network [MSH02].

The Department of the Army has specified the Land Warrior system to assist the small infantry unit with two tasks crucial to winning the close combat fight: (1) pre-arranging the conditions of the fight; and, (2) striking the enemy with a decisive maneuver. For the first task, the small unit leaders must have real time access to the all the information about the upcoming battle in order to develop the situation while out of contact with the enemy. A majority of the information is available from sources higher in the command structure and can include terrain information, obstacle information, and composition and disposition of enemy and friendly forces. The primary requirement of the communications system for pre-arranging the conditions of battle is the rapid, efficient, and correct assimilation of information from multiple sources in different formats [ORD01].

Just as the first task depends on information transfer and assimilation for a small unit to develop a plan allowing them to create the best conditions for mission success, the second task also depends upon real-time information access to allow the unit to maintain superior situational awareness when striking the enemy. A majority of this information comes from the soldiers actively engaged with enemy forces. The unit leader's access to the information developed while in enemy contact allows him to decisively maneuver his subordinate units to positions maximizing their advantage over the enemy [ORD01].

Table 1 shows some sample requirements for metrics such as quality of service (QoS) and throughput by which we can determine if an enhancement to the 802.11 MAC layer meets a minimum standard of acceptability.

Task Name	Process Artifacts supporting task	Data Type	Typical size (Kbytes)	QoS Requirement	Throughput Requirement
(1) Pre-arranging conditions of the fight	1.1 Virtual Meeting 1.2 Overlay 1.3 OPORD 1.4 Positional	1.1 VoIP 1.2 Graphic 1.3 Email 1.4 GPS	1.1 500 1.2 750 1.3 500 1.4 20	1.1 300 ms 1.2 300 ms 1.3 300 ms 1.4 300 ms	1.1 750kB 1.2 750kB 1.3 750kB 1.4 750kB
(2) Striking enemy with decisive maneuver	2.1 Virtual Meeting 2.2 Overlay 2.3 Intel reports 2.4 Positional	2.1 VoIP 2.2 Graphic 2.3 Email 2.4 GPS	2.1 500 2.2 750 2.3 500 2.4 20	2.1 300 ms 2.2 300 ms 2.3 300 ms 2.4 300 ms	2.1 750kB 2.2 750kB 2.3 750kB 2.4 750kB

Table 1. Objective criteria requirements.

1.2 Wireless Local Area Networks

Wireless networks, WLANs in particular, are becoming an increasingly necessary part of any communications network topology to support the tasks defined above. The mobility and flexibility advantages of a wireless network are especially suited to the needs of a fighting force. First, the soldiers in the field benefit via increased mobility while remaining plugged into the network. Second, the flexibility of an ad-hoc wireless network allows the soldiers to quickly “task-organize”, where a mission commander will organize elements from different subordinate units into a force with the required resources to execute a particular mission. Flexibility of ad-hoc connectivity also supports final briefing meetings among subordinate unit leaders and their commander before a battle. The nature of data traffic and throughput requirements for the battlefield, in addition to the proximity of the fighting force members, makes WLAN a choice being considered for linking members of a ground-based fighting force in ad-hoc, dynamic configurations. [HSM02][ORD01]

Given that the Land Warrior system is based on WLAN technology, this WLAN supporting an infantry fighting force must be able to support real-time information access required by all soldiers in the force at acceptable levels of service throughput and reliability. Traffic can include graphic intensive map overlays, voice transmissions, operations orders, soldier position information, soldier status, equipment status, and intelligence reports. All of this traffic must be correctly and efficiently delivered through the network despite topology changes caused by interference, congestion, or lost nodes [NOP00]. The Army is also interested in having the wireless station nodes themselves being able to adapt to topology changes without human intervention (i.e. without requiring extensive network administrative function performed by a human) [NOP00].

However, the 802.11 protocols currently do not address the actions a node can automatically undertake in the face of dynamic environmental change or in the face of changing resource constraints. Currently, an 802.11b wireless node operates in one of two preset modes, distributed coordination function (DCF), which mostly used in ad-hoc WLAN configurations, and point coordination function (PCF), which provides contention free access by using a central controller node (i.e. a wireless access point) to arbitrate the communication medium [GAS02].

Both modes control access to the wireless medium using differing access control techniques. The DCF uses carrier sense, multiple access with collision avoidance (CSMA/CA) to control which station may access the medium and, with spacing delays between attempted transmissions, lock out the medium when it is in use. The PCF is used to force a “fair” access to the medium by allowing some stations a shorter interval between transmissions. Neither of these functions allows a node to automatically change from one function to another in the face of resource constraints such as bandwidth congestion or storage constraints, for example [GAS02]. Human intervention in the form of network management is required to force this kind of change.

Currently, the radios used in the Land Warrior system use the On Demand Multicast Routing Protocol (ODMRP) to account for loss of line of sight between radios. The ODMRP sits on top of the 802.11 protocol stack and routes packets to multiple stations. These packets can time-out or be discarded when they fail to reach their intended destination [MSH02]. The protocol builds a mesh of shortest paths between members in the group by having nodes periodically broadcast a "member advertising packet", which updates the route from the source node to all other members in the group. Even though this protocol is meant to reduce the overhead associated with similar *unicast* protocols, it still requires some overhead to build and maintain the routes between nodes [LSG00].

1.3 Modeling the Wireless Battlefield Using the *ns-2* Network Simulator

In this thesis, we will use simulation as a method for creating fighting force network models, to be used in evaluating a set of possible extensions to the 802.11b MAC layer, with the research objective of laying the groundwork for studying how this protocol could be extended via adaptive agents techniques to become more self-regulating. Our research interest, and the contribution we seek to make, is in the area of combining conceptual modeling of the battlefield domain with the model creation activities required to create a viable and accurate networking simulation model of the communication s infrastructure of the fighting force on the battlefield.

In general, modeling and simulation gives us the opportunity to extend and test, under varying conditions, a networking protocol layer contained in a communications infrastructure without the cost of building a live infrastructure for testing. Without simulation, it has been noted in the literature that most communications hardware units can end up discarded due to failure [BRE00]. We anticipate the simulation models we create will allow protocol enhancements to be discovered, subsequently leading to adaptive wireless protocols to be built in an appropriate hardware delivery platform. The model development and simulation work covered under the scope of this thesis is an important first step in the process of developing a new, or extending an existing, network protocol.

More to the point, the *ns-2* simulation engine offers several advantages we will exploit for our model creation and validation via experiments. First, *ns-2* offers a varying amount of abstraction that will allow us to examine the 802.11b MAC protocol at a level conducive to our experiment—namely, considering the behavior of the medium management without considering the details of the protocol layers above and below the MAC layer.

Second, we can develop different scenarios that account for different dynamic properties of the domain model, such as: (1) node movement following the patterns of troop movement, (2) topology changes that are consistent with battlefield planning scenarios, (3) varying traffic patterns and volume that follow the time-varying nature and mix of data and voice traffic on the battlefield, and, (4) dynamic events that perturb the networking infrastructure. These are all to be explored as a means to test the network model created under situations as close to battlefield conditions as possible.

Finally, it is possible to test an extension to the 802.11b MAC protocol by changing the definition of the wireless LAN model by modifying the source code of the simulator to fit our needs, then run the scenarios again [KFV00]. These qualities make *ns-2* a useful tool for simulation studies of this nature. It is the objective of this research to make it possible to easily extend the domain model and the corresponding simulation model, so that the ASOWN research team can carry out the extension of protocol capabilities as needed.

1.4 Statement of Research Problem

The purpose of this research is to create a modeling and simulation model and methodology, based on the use of the Unified Modeling Language for the conceptual analysis of the battlefield domain, and the use of the *ns-2* simulator for creating operational models for simulating the WLAN infrastructure on which model experiments can be executed. This work has been undertaken as part of the larger ASOWN project, which has as its objective to make the 802.11b MAC layer wireless protocol more robust and flexible for use in a battlefield environment by focusing on use of adaptive, self-organizing agents for resource management at this layer of the network.

The ASOWN project seeks to make the 802.11 protocols better at self-regulation to alleviate the need for human intervention to regulate and manage the network. The literature points to use of agent-based techniques for network management, advocating their use in making changes to a network after a topology change or the onset of congestion in lieu of having a “human in the loop”. In addition, the automated network management of the MAC layer would not be subject to the stress of battlefield conditions that human soldiers experience. It is a tacit assumption of this research that knowledge exists at the MAC layer that is sufficient to create a set of adaptive behaviors to identify patterns, for example, to determine if resources are becoming constrained, or if a topology change has occurred affecting individual node performance or performance of the overall network; it is assumed that—once such patterns are identified via analysis, and validated as being "interesting" via simulation, the individual nodes can determine and force some changes in the network automatically.

For example, one node could be acting in PCF mode as an access point. If this node is required to store packets for another node that has powered down or moved out of range, the first node could start losing memory space. In that case, it may poll the rest of the network for another node that can store the packets until the node in question returns to service. In another scenario, a node operating in DCF mode could detect congestion in the network by the number of retries it is required to make before a successful transmission. In this case, the node could elevate itself to PCF mode in order to set a priority on the traffic in order to relieve the congestion.

1.5 Statement of Solution Approach

We will investigate plausible solution architectures by developing an experimental framework for exploring a set of changes to the 802.11b MAC protocol and by simulating a battlefield wireless network on the *ns-2* simulator platform. The *ns-2* simulator allow one to try different sets of protocol architecture extensions to the MAC layer by making changes to the simulation models that would be impractical with real WLAN equipment. In the simulator, one can run scenarios of an infantry platoon performing a set of missions on the battlefield. However, before such a simulation model can be created, analysis of the fighting force domain must be carried out.

Scenarios that depict a fully functioning WLAN network can provide a baseline for metrics such as quality of service (as measured by delivery time), throughput (frames per second), percentage of frame retries (indicating a level of congestion), and recovery time (mean time to repair), as well as providing data on the impact of human response time on network resilience and level of service. By creating running simulation scenarios that model changes in the WLAN, simulating battlefield conditions and the movement of the fighting force in that environment, we can measure how much traffic is lost and the time required to return to acceptable levels of service and throughput in the networking environment.

Some network changes will be based on troop movement, which increases or decreases the distance between nodes or affects the direction of node movement in relation to one another. Other network changes will be event based, which could result in eliminating a WLAN node, either as a casualty of enemy fire, a node powering down, or a node moving out of range. We will seek to introduce event-based network changes at different levels of traffic intensity and patterns of movement to see the impact on network resilience.

We will also establish a broader baseline by measuring how the standard 802.11 protocols react to the changes in the status of individual or collection of nodes. This allows us to see how a set of enhancements to the MAC layer protocol will increase or decrease the ability of the network to self-regulate and quickly return to an acceptable level of performance under different operating conditions.

1.6 Statement of Research Method

The following activities constitute the research method undertaken as part of this research, documented in this thesis:

1. Define and develop the scenarios representing the fighting force: These scenarios will be based on an infantry platoon and subordinate units conducting a set of standard missions on a battlefield. In addition, they will be based on assumptions about network traffic loading, types of traffic, and assumptions regarding quality of service—built from a careful analysis of the domain, using UML as the medium for initially capturing the salient semantic properties of the domain.
2. Develop the simulation model of the fighting force's wireless communications infrastructure: The simulation model instantiates and/or extends the MAC baseline protocol models being used in *ns-2* for executing the baseline fighting force movement scenarios: The model should account for the formations and patterns of movement of the fighting force (and, therefore, the "wearable" computing and communications infrastructure), as part of the baseline modeling input into *ns-2* experiments. The creation of an appropriate simulation model is based on the results of the domain analysis, and the research activity involves identifying a mapping transformation between domain model and simulation model

3. Test the baseline model. We will run all scenarios using the 802.11 protocol extensions to *ns-2* in order to establish a baseline for our metrics and define the optimality criteria. This data will be organized and statistics will be plotted.
4. Codify the process of using this method for exploring the 802.11-based wireless network infrastructure for a fighting force: We will document the process of mapping between domain modeling and simulation modeling and analysis—which are very different meta-domains with very different underlying modeling methods and principles

1.7 Thesis Outline

We document the results of this research effort in the following manner. In Chapter 2, we will characterize the model of the fighting force domain—an activity that, to our knowledge, has not been done using conceptual modeling techniques and notations. Chapter 3 will discuss the relevant aspects of the current 802.11 protocols and the use of simulation in *ns-2*, and the appropriateness of this endeavor to the military problem under consideration. Chapter 4 will present the fighting force enterprise model created through the conceptual modeling activities, using UML as the representation medium. Chapter 5 will present the simulation model developed as a result of the domain analysis and the mapping of domain to simulation vocabulary and ontology. Chapter 6 will present the defined metrics, the design of the experiments, and the results of the experiments. Chapter 7 will analyze our conclusions, pointing to the efficacy of this model for use as a basis for more extensive WLAN infrastructure modeling.

CHAPTER 2

DEPICTION OF THE BATTLEFIELD DOMAIN

2.1 The Infantry Platoon

In order to create our analysis model from which we can devise a set of simulation scenarios to explore and evaluate the IEE 802.11b MAC network being adopted, we chose a U.S. Army light infantry platoon as the domain. The choice of infantry was easy since the Land Warrior system is intended to be the primary means of communication for infantry units from fire team to division level. We chose an infantry *platoon* because it is the primary means for the Army to take and control terrain [SHI99]. It also provides a fair representation of Army organizational unit, which we can easily use to scale to analyze and model larger units, when needed.

In addition, the light infantry platoon is capable of performing the full range of missions required by the U.S. Army using a full complement of tactical formations and movement techniques. The platoon uses the standard techniques for mission preparation and execution used by all other Army units. Finally, the platoon provides enough network traffic to explore how the network reacts to different network dynamic events.

We will describe how the platoon fits within the structure of the Army, its unit organization, the different missions it can perform, the different unit movement tactics it uses, a general format for mission execution and how the different stages in mission execution affect network traffic, and give a general description of the mission we used for this model.

2.2 High-Level Actors

The U.S. Army, while encompassing several million people on Active Duty and in the Army Reserve or National Guard, can be modeled at a high level of abstraction by three course actors: Command, the Fighting Force, and the Enemy Force (Figure 2-1). The Command actor issue mission orders, the Fighting Force carries out those orders, and the Enemy Force resists the completion of those missions. On the friendly side, how we label a unit depends upon our level of abstraction. We will define the Fighting Force *actor* as the unit at our current level of abstraction. Therefore, the Command actor will be the unit just above our level of abstraction. For example, at the Fire Team level, the Fighting Force is the Fire Team and the Command is the Squad. At Squad level, the Fighting Force is the Squad and the Command is the Platoon. This type of labeling can be applied at any unit level within the Army hierarchy. However, it is very simplistic. Therefore, a detailed explanation of an infantry platoon organization is required.

2.3 Modeling the Infantry Platoon

The infantry platoon is a three-level hierarchy (Figure 2-2). At the lowest level is the fire team (Figure 2-3). A fire team consists of four soldiers, the Team Leader (TL), Grenadier (GNDR), Automatic Rifleman (AR), and Rifleman (RM). The GNDR, AR, and RM are all young soldiers who are responsible for maintaining themselves and their equipment. The different labels are derived from the different weapons these soldiers carry. The TL maintains control of his team, controls their movement and rate of fire, and maintains accountability of all soldiers and equipment within the team. The fire team is the lowest maneuver element within the infantry and generally is the first unit to make contact with enemy forces. The next level up is the squad.

A light infantry squad consists of two fire teams and a squad leader. The squad leader trains the squad on combat tasks, manages the squad's logistical and administrative needs, directs maintenance on equipment and weapons, and maintains accountability of the soldiers and equipment in his squad. The next level up is the platoon.

A light infantry platoon has three squads and a headquarters (HQ) team. The platoon leadership, the forward observer, the radio/telephone operators, and two machine gun crews comprise the HQ team. A platoon leader (PL) and a platoon sergeant (PSG) combine to lead the platoon. The platoon leader (PL) commands the platoon and is primarily responsible for accomplishing the platoon's tactical mission. He develops and ensures execution of the tactical plan, going anywhere needed within the platoon to ensure mission success. The Platoon Sergeant (PSG), the platoon's senior Non-Commissioned Officer (NCO), ensures the platoon is trained to the Army standard, directs the platoon Aidman, supervises the platoon's supply, support services, and administrative effort, and provide expertise and experience in the execution of the platoon's mission. The PSG can also take charge of platoon elements detached to perform a specific task required to accomplish the platoon's mission. The Aidman monitors the health and hygiene of member within the platoon and, under the control of the PSG, treats and evacuates casualties. The HQ team also has a Fire Support team consisting of the Forward Observer (FO) and the FO Radio/Telephone Operator (FO RTO). The FO works for the PL and develops a fire support plan for each mission in order to support the platoon with indirect fire. During execution, the FO will call for and adjust fire as needed by the situation. The FO RTO establishes, operates and maintains the communications equipment for the FO. If needed, the FO RTO can perform the FO's duties. The PL also has an RTO who installs, operates, and maintains the communications networks for the platoon. Finally, the HQ team has two Machine Gun (MG) Crews, each consisting of a gunner and assistant gunner, who support the platoon's mission with heavy weapons direct fire. The PL will position the MG crews as needed to best support the platoon's mission. As you can see, the light infantry platoon is a hierarchical organization built upon teams of soldiers. Given the organization of the platoon, we will now describe how the platoon moves on the battlefield.

2.4 Movement

Movement is one of three tactical operations undertaken by an infantry platoon. Understanding how the platoon moves is essential to understanding its effect on the wireless network. The PL must make two choices when deciding how the platoon to move towards its objective. First, he must decide on a formation, or how to arrange the squads, fire teams, and soldiers in relation to one another. Second, he must decide on a movement technique, or how the squads and fire teams are arranged in relation to each other during movement. Movement techniques and formation differ in that formations are relatively fixed, while movement techniques are not. Formations are designed to allow the platoon to weight it's firepower in a particular direction. Movement techniques allow the platoon to make contact with the enemy with the smallest element possible, allowing the rest of the platoon to maneuver to a more advantageous position. We will first describe the different formations.

2.4.1 Formations

At the platoon level, the PL has six formations from which to choose. The PL decides which formation to use based on how well the characteristics of a particular formation meet the requirements of the mission in the areas of control, flexibility, fire capabilities or restrictions, and security. Control describes the ease with which soldiers can see their immediate leader and how easy it is for the leaders to issue non-verbal commands to control their unit. Flexibility describes the ease or difficulty with which a unit transitions from one formation to another. Formations also provide different fire capabilities, which describe how the leader can weight the unit's firepower in the direction of expected enemy contact. Finally, formations allow differing amounts of security by allowing the unit leader to orient soldiers to look all around the formation or to only look in one direction for possible enemy contact. Each formation is built upon formations at the fire team level.

The fire team uses two basic formations: the wedge and the file. The wedge is shown in Figure 2. Depending upon the position of the RM, it can be either a wedge left or wedge right. The choice of which to use depends upon where the leader decides to have most of the fire team's firepower.

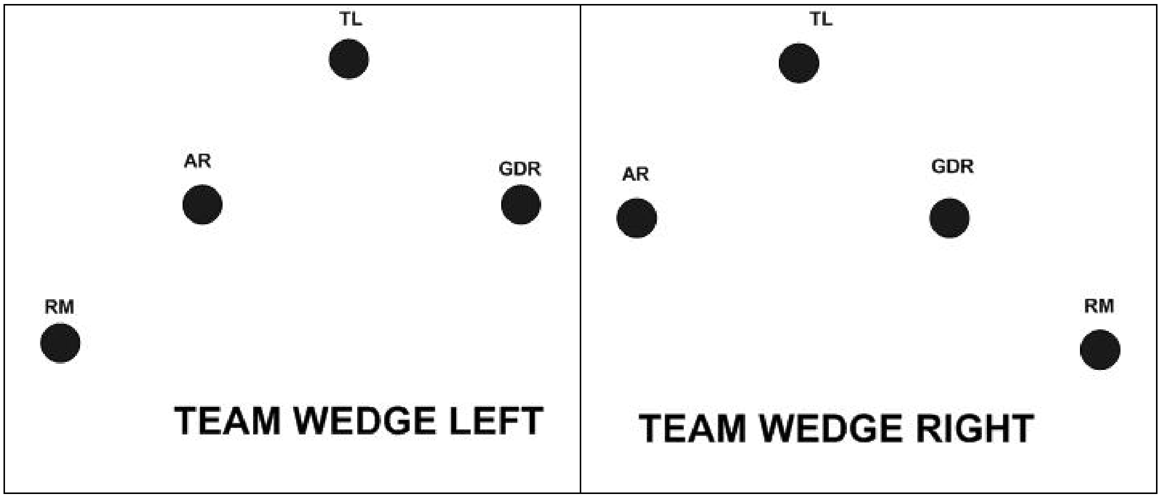


Figure 2. Fighting force formations for the team unit.

The next basic fire team formation is the file, shown in Figure 3. It is generally used at times when limited visibility, as caused by light conditions, weather, close terrain, or dense vegetation, precludes use of the team wedge.

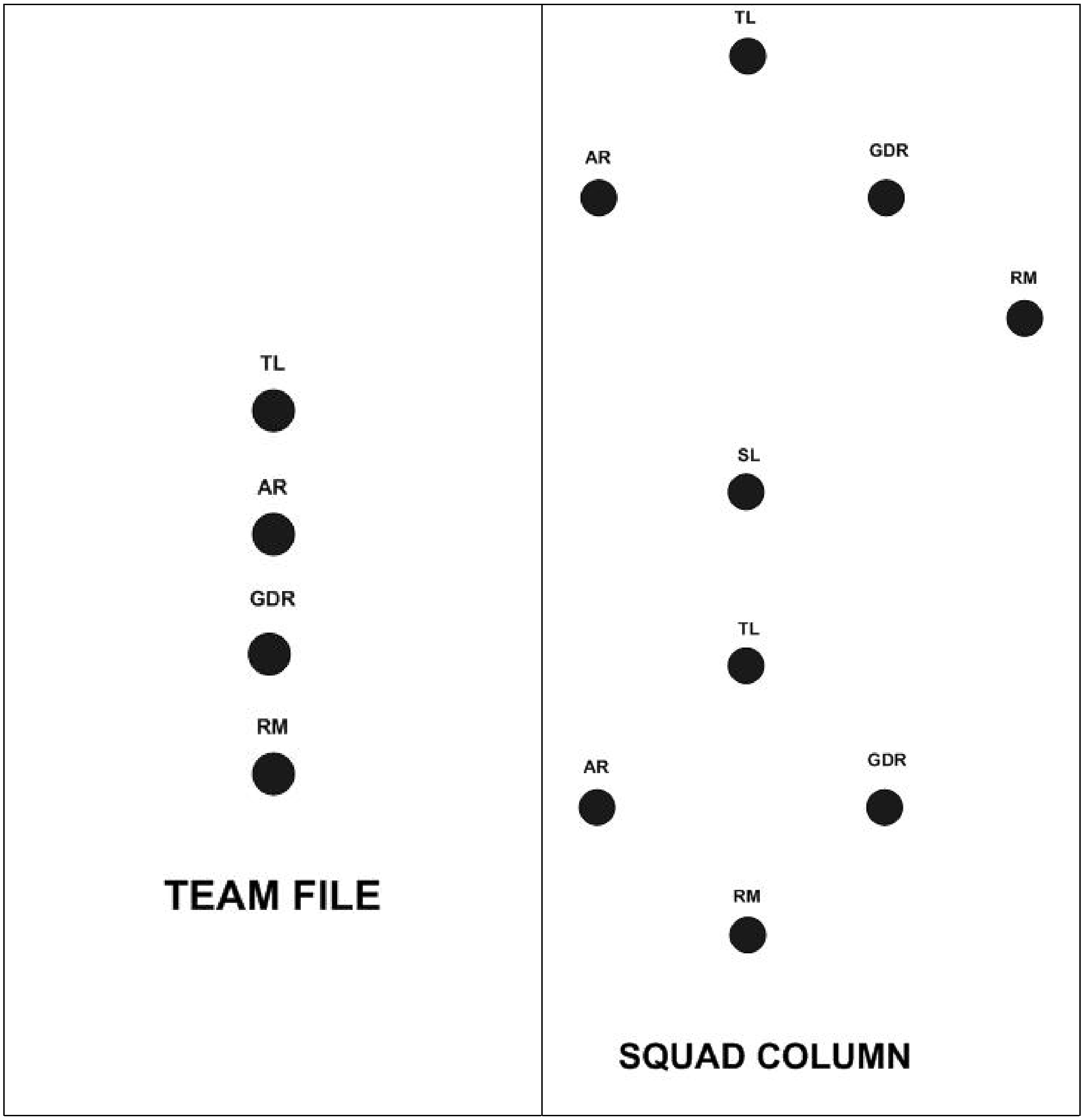


Figure 3. Fighting force formations for team and squad.

Building upon the fire team formations, the squad has three formations that describe the relationship between fire teams. The most common squad formation is the squad column as shown in Figure 3. It provides the best mix of control, flexibility, fire capability, and security, especially when the enemy situation is vague.

The next squad formation is the squad line (Figure 4). It is primarily used when maximum firepower is required to the front of the formation.

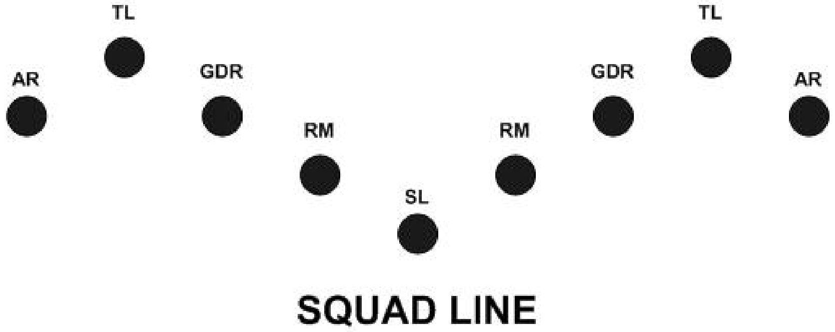


Figure 4. Squad line formation of the fighting force.

The final squad formation is the squad file (Figure 5). It is used in limited visibility situations since it provides the best control of the three-squad formations. The three-squad formations are the building blocks for the platoon formations.

As with the team wedge and squad column, the platoon column is the platoon's primary formation. It provides the best mix of control, flexibility, fire capability and movement to meet most battlefield situations. The platoon column is also shown in Figure 5.

The next formation is the platoon line, squads-on-line formation, shown in Figure 6. It provides maximum firepower to the front and is the platoon's primary formation from which they will launch an assault on an objective. When using this formation, the PL normally knows how the enemy is arrayed on the battlefield.

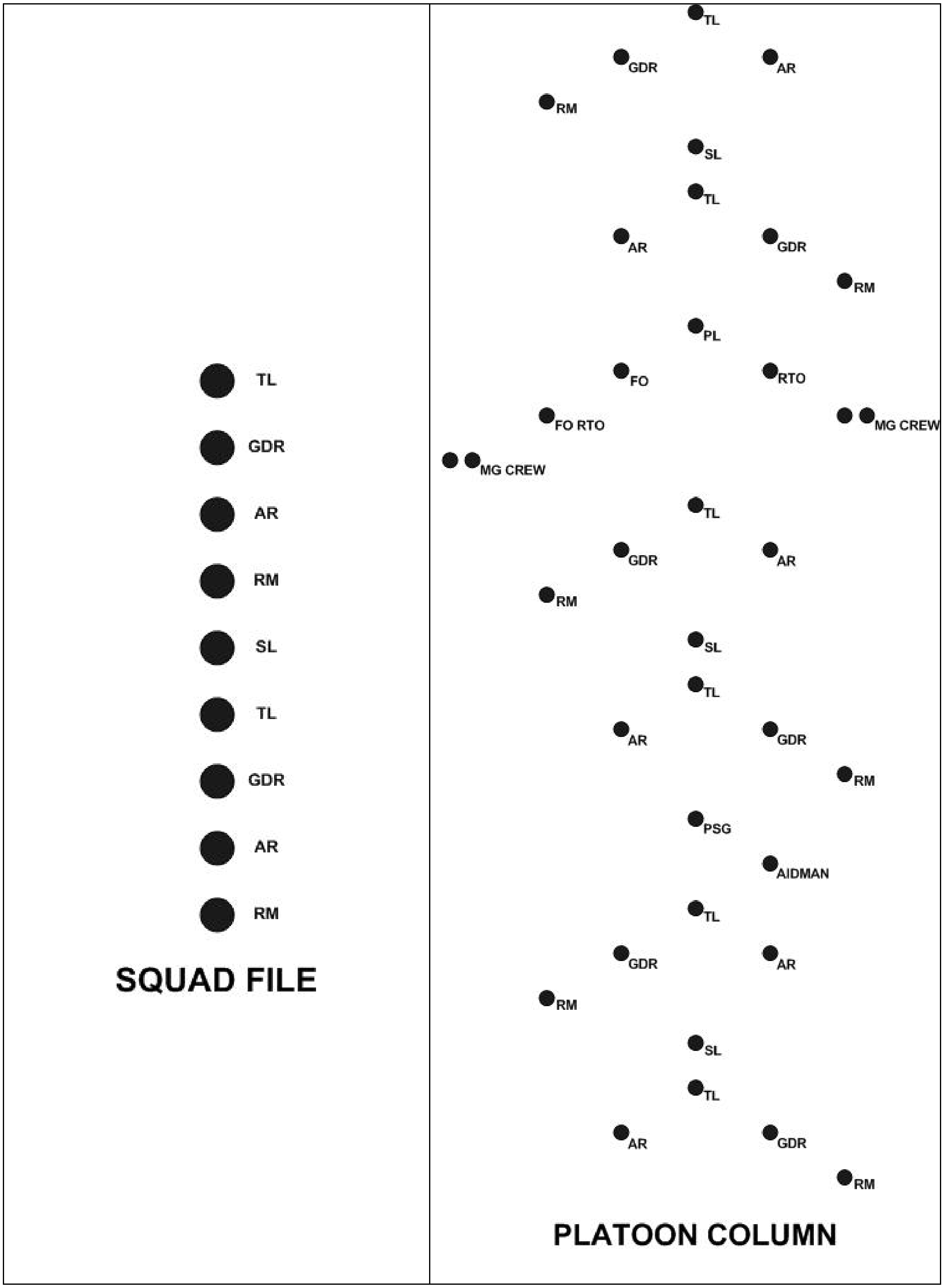
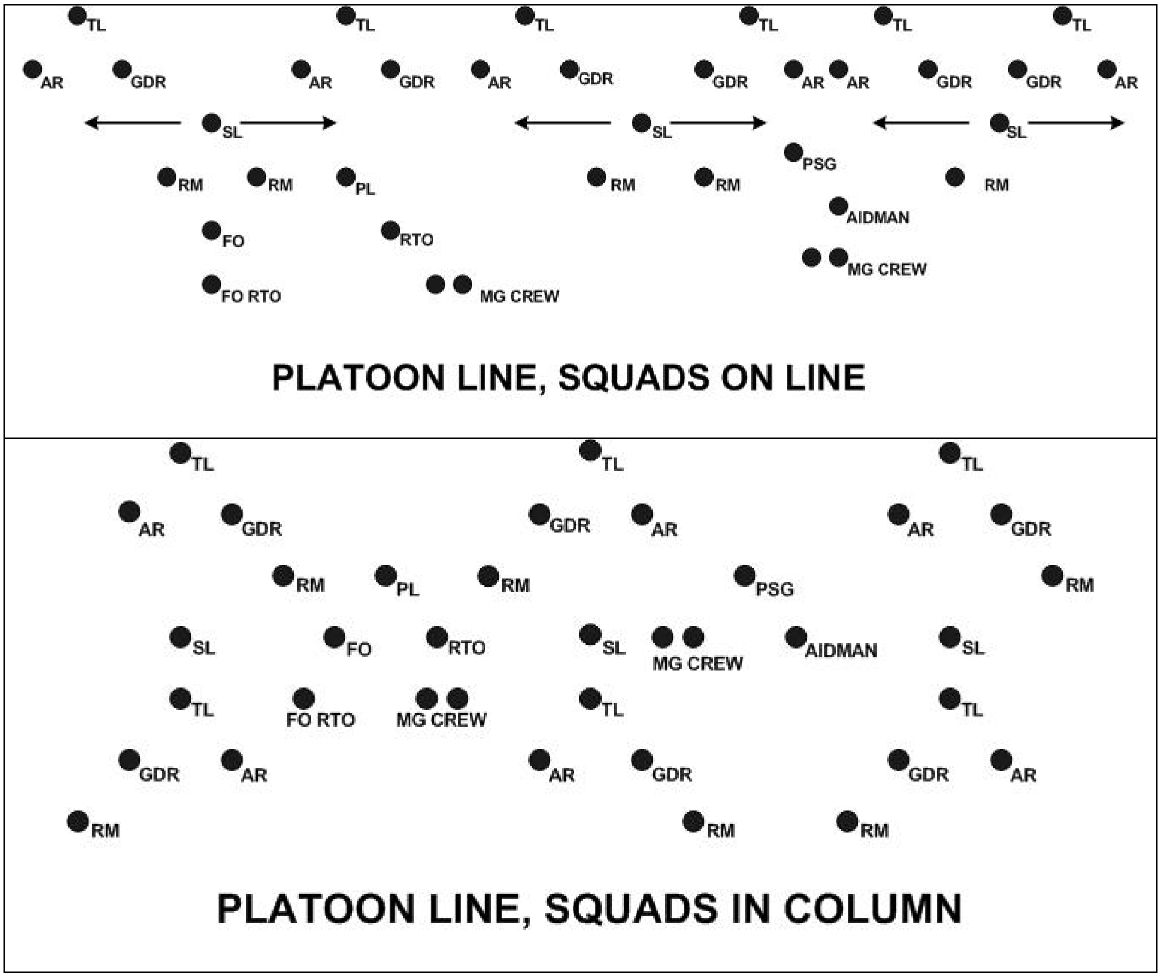


Figure 5. Squad and platoon formations of the fighting force.



PLATOON LINE, SQUADS ON LINE

PLATOON LINE, SQUADS IN COLUMN

Figure 6. Platoon line and column formations of the fighting force.

The platoon line, squads-in-column formation (Figure 6), is similar to the platoon line, squads-on-line, but is used when the PL does not want all soldiers at the front. This formation allows the PL a greater degree of flexibility than the platoon line, squads-on-line for situations when the enemy situation is not as well known.

The platoon wedge of Figure 7 and the "vee" of Figure 8 are both used when the enemy situation is vague, but with one important difference. The PL will use the platoon "vee" when he expects contact from the front and the platoon wedge when contact is not expected. The wedge allows the PL more flexibility than the “vee” by allowing him to make contact with a small element, develops the situation, and still has two squads to maneuver against the threat.

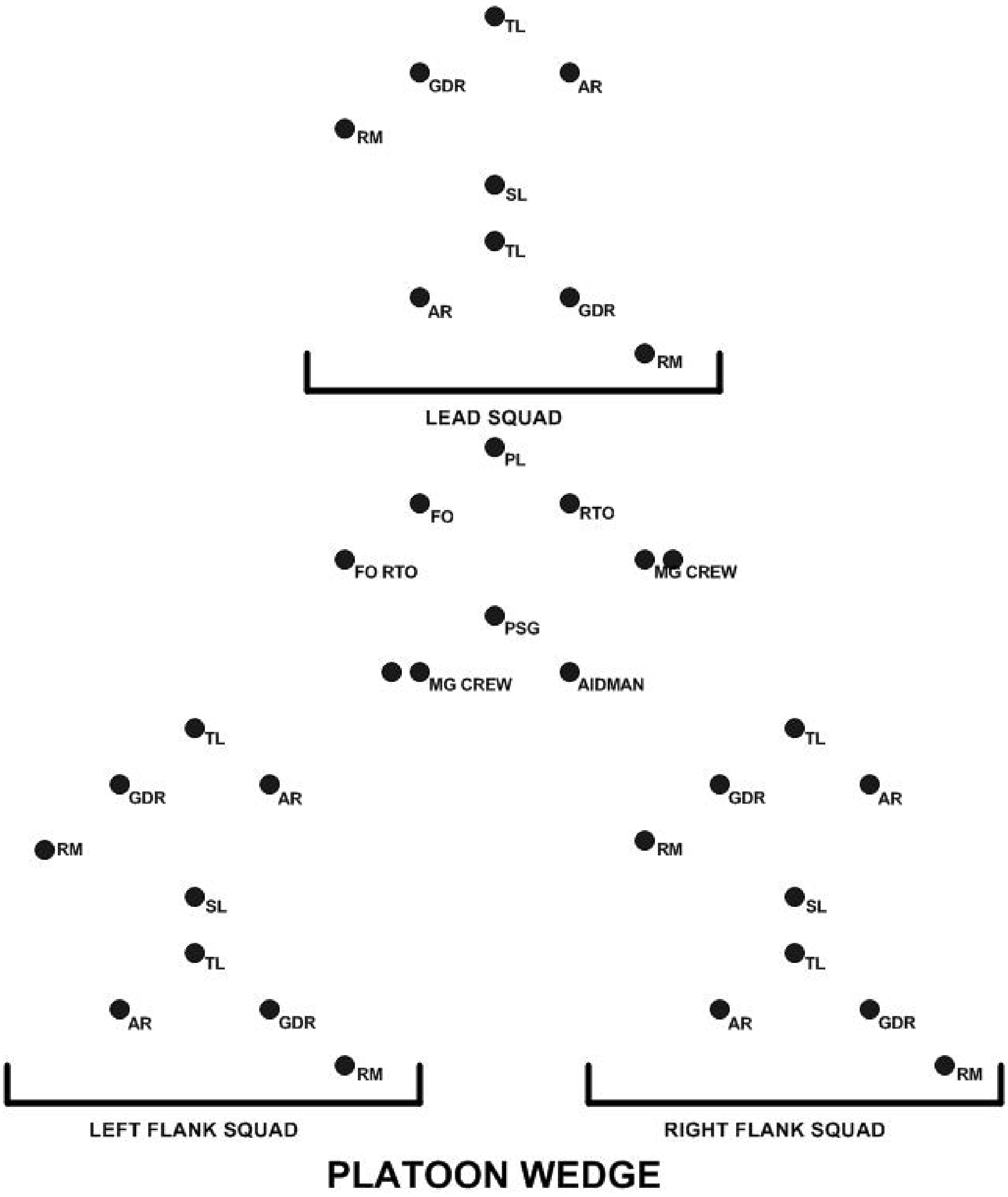


Figure 7. Platoon wedge formation of the fighting force.

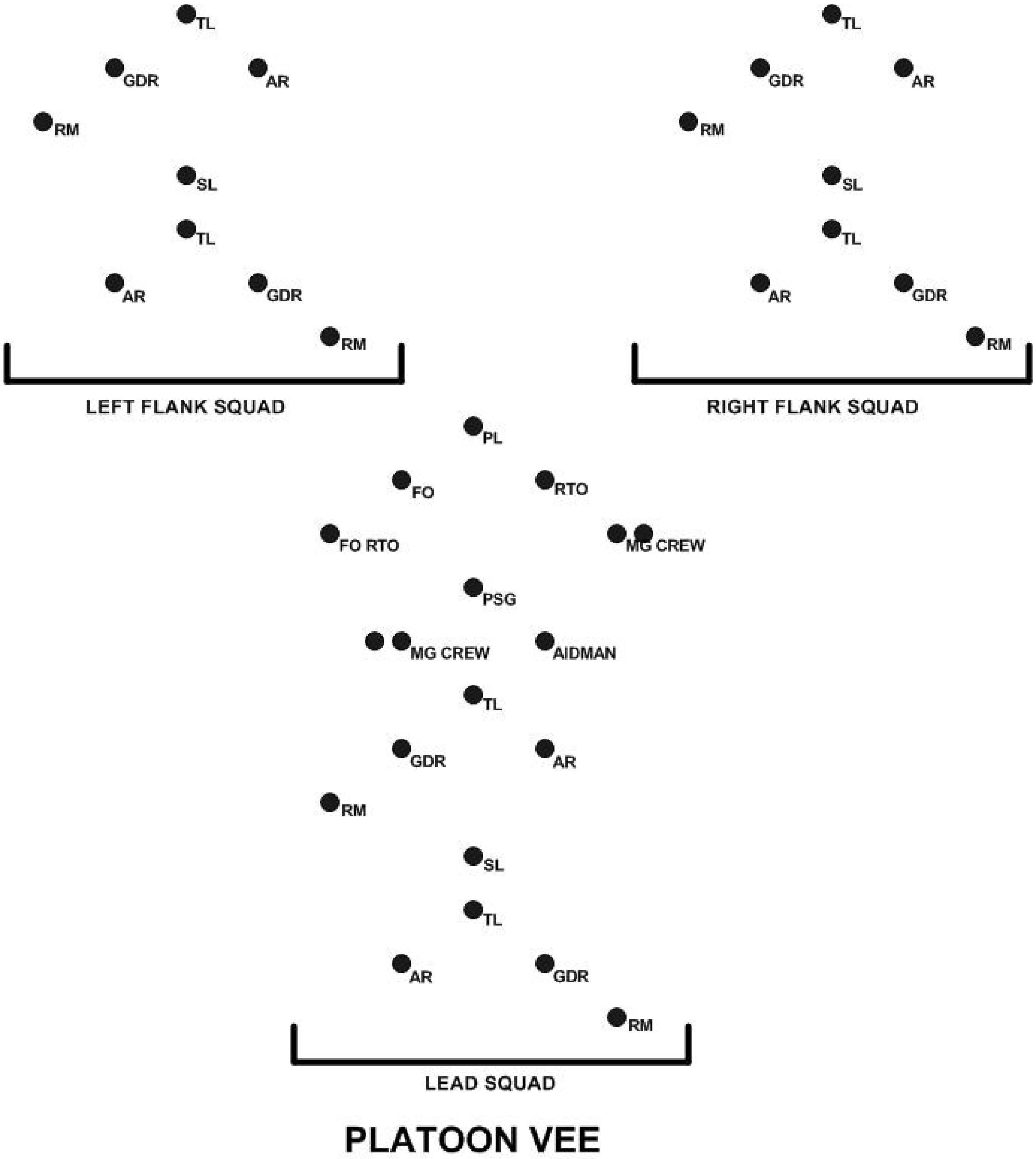


Figure 8. Platoon "vee" formation of the fighting force.

The final platoon formation is the file. As with the team file and squad file, this formation is used during periods of low visibility. It also has the same characteristics as the other files. However, the PL can choose two variations. One is a simple file of three squads. The other has a point and two flank security teams, as shown in Figure 9.

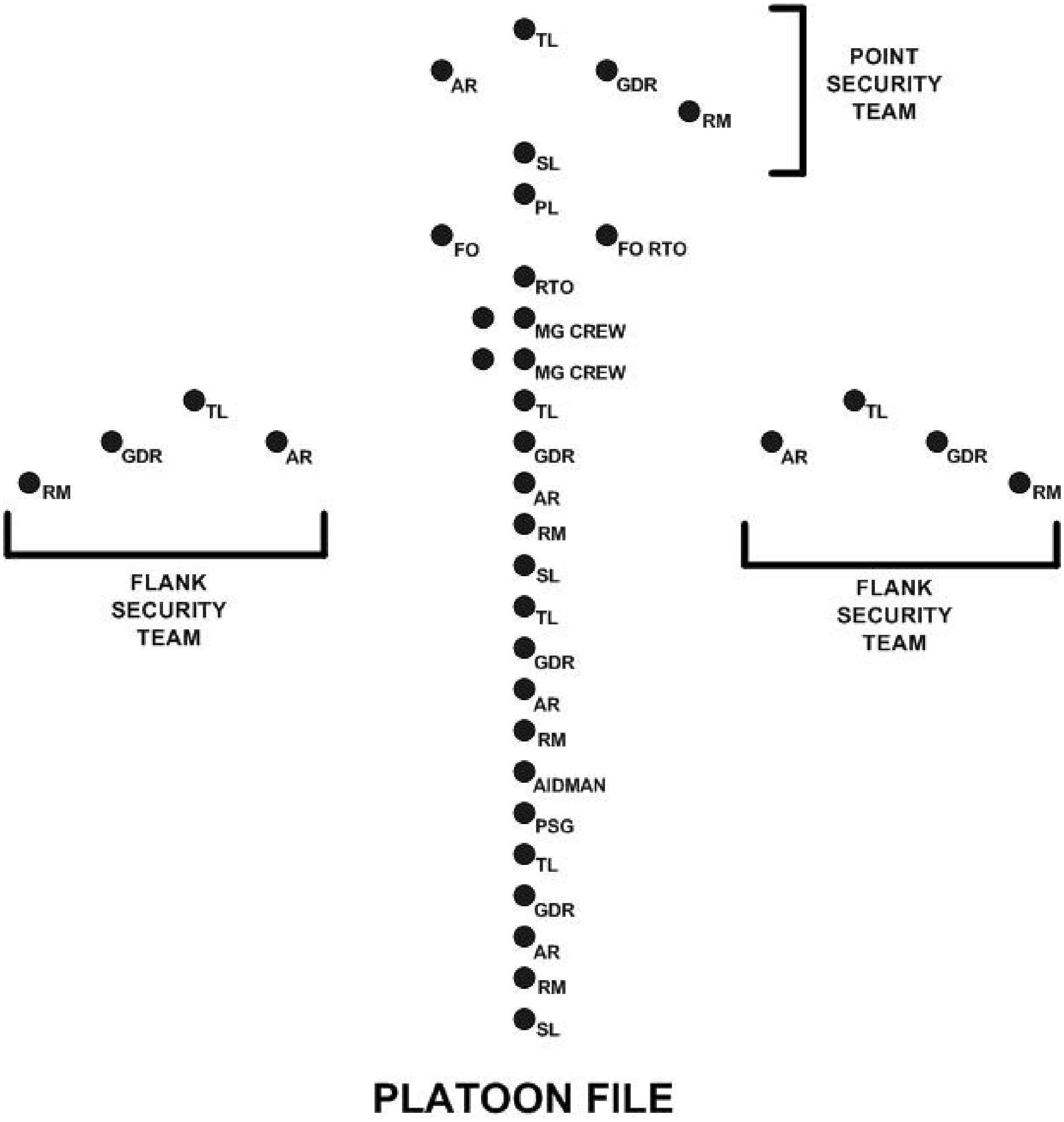


Figure 9. Platoon file formation of the fighting force.

2.4.2 Movement Techniques

In addition to deciding which formation to use, the PL must also decide which movement technique to use during each phase of the mission. The movement technique employed changes during the course of the mission because the factors governing their use changes. The PL weighs the likelihood of enemy contact and how well the characteristics of control, dispersion, speed, and security for each technique meet the needs of the situation. Control describes how well the leadership can effectively disseminate commands to the rest of the unit; dispersion, the distance between soldiers to guard against weapons with an area effect; speed, how fast the unit can move across the battlefield; and security, how well the unit can guard against being surprised by the enemy. Table 2 shows the relative rankings of each movement technique for each characteristic. Any of the movement techniques can be used with any formation. The different distances between subordinate elements describe the difference between the techniques.

Movement Techniques	**Enemy Situation**	**Control**	**Dispersion**	**Speed**	**Security**
Traveling	Contact Not Likely	More	Less	Fastest	Least
Traveling Overwatch	Contact Possible	Less	More	Slower	More
Bounding Overwatch	Contact Expected	Most	Most	Slowest	Most

Table 2. Ranking of characteristics and movement techniques.

Both the squad and platoon use the same movement techniques, but the meaning is slightly different at each level. For the squad, traveling requires approximately 20 meters between the lead and trail squad (Figure 10). For the platoon, it requires approximately between each element within the platoon (Figure 11). With *traveling overwatch*, the squad increases the distance between the teams to 50 meters (Figure 12). In the platoon, the first squad will use *traveling overwatch* and increase its distance from the rest of the platoon to 50 meters. The trailing squads continue to use the traveling technique (Figure 13).

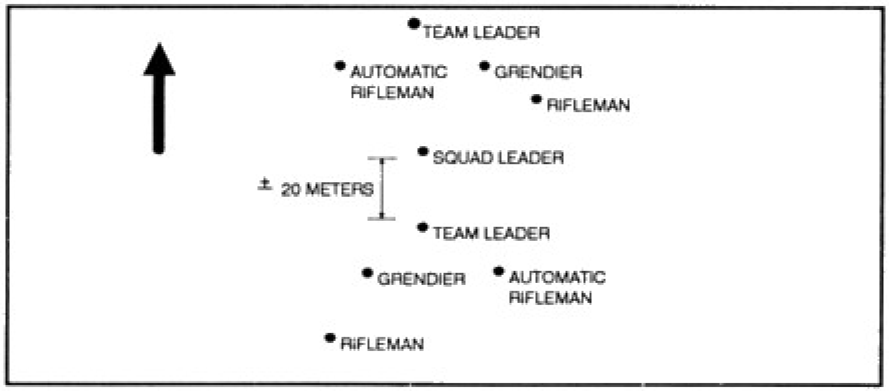


Figure 10. Platoon file formation of the fighting force.

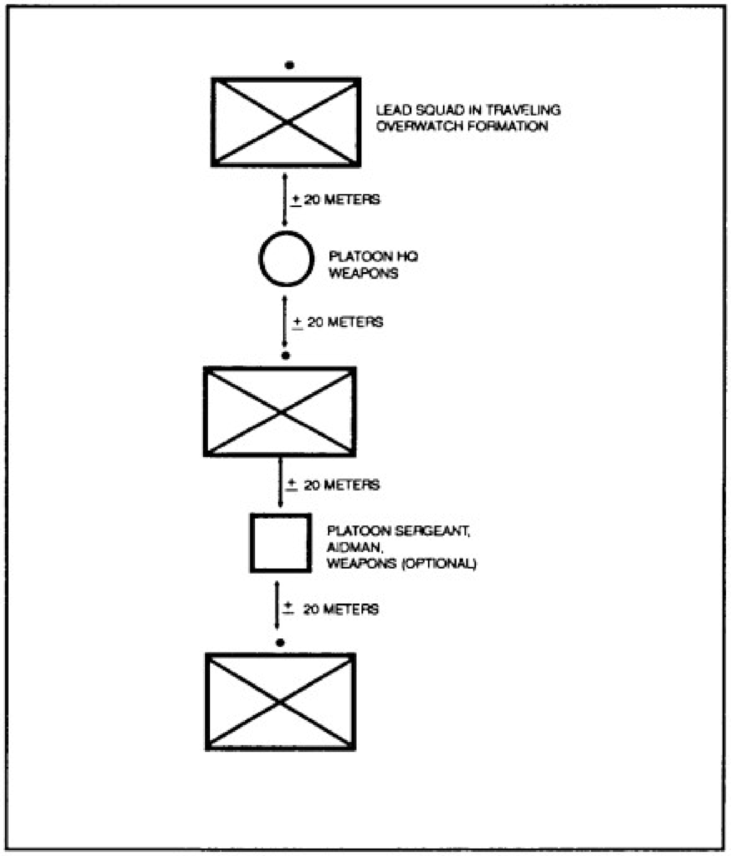

LEAD SQUAD IN TRAVELING OVERWATCH FORMATION
± 20 METERS
PLATOON HQ WEAPONS
± 20 METERS
± 20 METERS
PLATOON SERGEANT, AIDMAN, WEAPONS (OPTIONAL)
± 20 METERS

Figure 11. Platoon file formation of the fighting force.

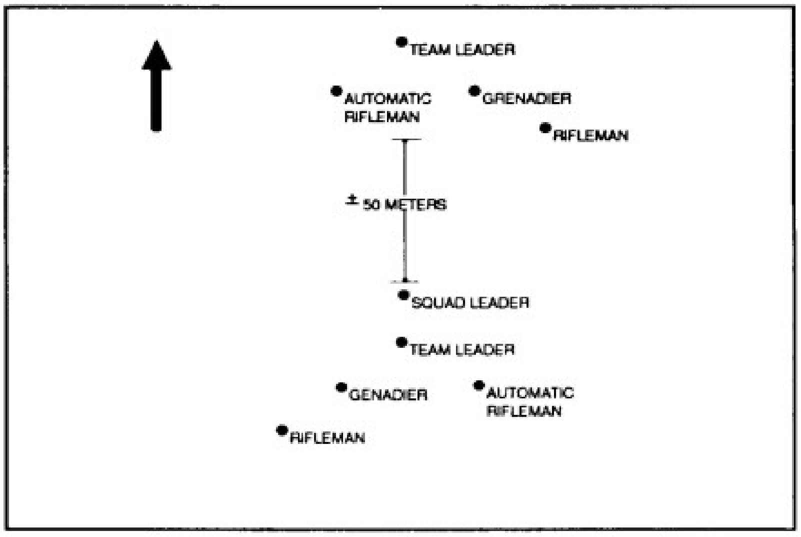


Figure 12. Platoon file formation of the fighting force.

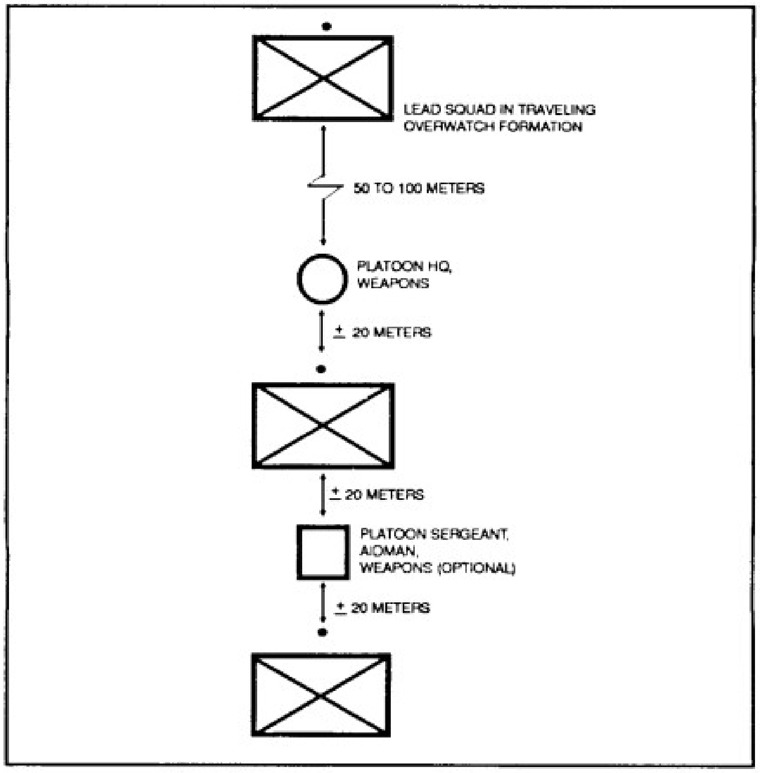

LEAD SQUAD IN TRAVELING OVERWATCH FORMATION
50 TO 100 METERS
PLATOON HQ, WEAPONS
± 20 METERS
± 20 METERS
PLATOON SERGEANT, AIDMAN, WEAPONS (OPTIONAL)
± 20 METERS

Figure 13. Platoon file formation of the fighting force.

Bounding overwatch is drastically different from the other two movement techniques and is used only when the leader feels or knows the enemy is near and contact is expected. At the squad level, the SL will position one team in an overwatch position where they can fire at the enemy if needed. The other team is instructed to move to a position closer to the enemy. Once the second team is in position, it "overwatches" for the first team as they bound to a new position. In this manner, the squad can watch the enemy while still making forward progress. Depending upon the need for speed, the squad can use either successive bounds or alternating bounds. Figure 14 shows examples of both types of techniques.

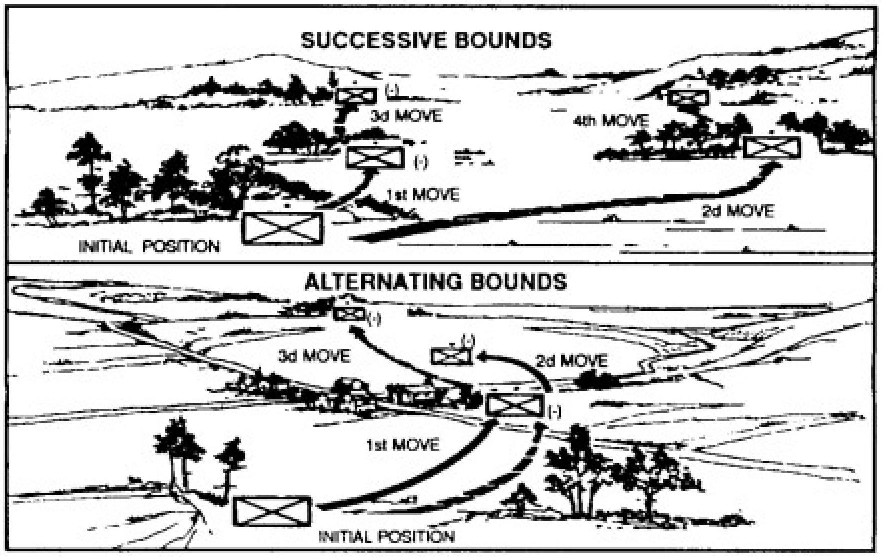


Figure 14. Platoon file formation of the fighting force.

Bounding overwatch for the platoon is similar to the squad, with the major exception being that the platoon has three subordinate elements as opposed to just two. During bounding overwatch, one squad will be in an overwatch position, one will “bound” to a new position, and the final squad will be awaiting or receiving orders. Figure 15 shows an example of bounding overwatch for the platoon.

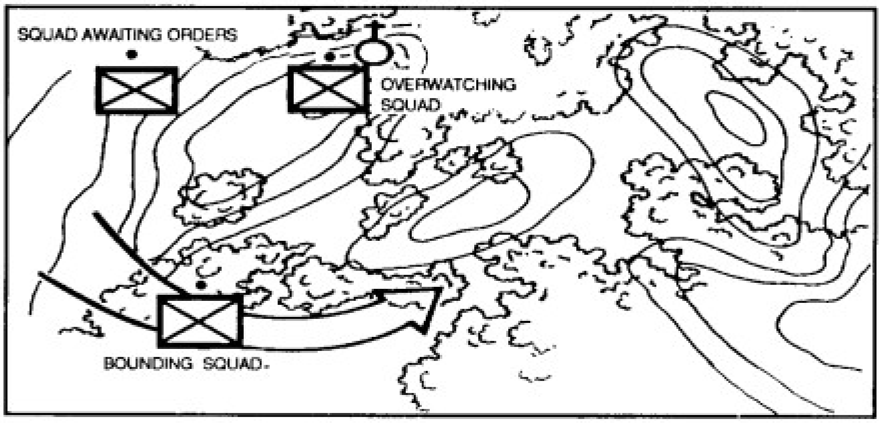


Figure 15. Squad formation of the fighting force.

2.5 Modeling Network Traffic

Unit formations and movement techniques help to show the spatial relationship between soldiers and, in the case of Land Warrior, the spatial relationship between wireless communications nodes. The mission preparation and execution process help to describe the type and volume of traffic on the wireless network. For the purposed of this thesis, we chose to focus on the mission execution portion of this process. At the platoon level and below, the mission execution provides most of the network traffic. In addition, most events that disrupt the network will occur during mission execution.

On an infantry platoon wireless network, network traffic is derived from four sources: electronic mail (email), graphical map overlays (standardized graphics that can be drawn on a topographical map to convey information such as unit locations and control measures to guide unit movement), global positioning system (GPS) data, and voice-over-IP (VOIP). All of the artifacts produced by the mission preparation and execution process fit within one of these four categories. Traffic from these sources can travel up and down the hierarchy and between soldiers within the same unit. From company level, platoons will receive artifacts such as operations orders (OPORDs), fragmentary orders (FRAGOs), situation reports (SITREPs), map overlays, and verbal commands. The soldiers in the teams, squads, and platoon will send up GPS data, SITREPs, enemy reports (SALUTE reports), request for information (RFIs, used to clarify orders), and contact reports.

Besides the amount of network traffic, network dynamic events also occur primarily during mission execution. Network dynamic events include formation changes, dropped links, and dropped nodes. Formations can change based on changes in the terrain or changes in the enemy situation. Soldiers moving behind “folds” in the terrain can interrupt network links, or movement behind dense vegetation can degrade it. Nodes can drop because of equipment failures or soldier casualties. An additional complication for the network is that the periods of time with the highest probability of network events that degrade the network are also the periods of time with the highest likelihood for network traffic.

In order to show the source of network traffic and network events, we have developed a use case diagram for an infantry attack mission as shown in Figure 14. In this particular diagram, the Fighting Force has received the OPORD and is either just about to start or has just started the mission. This diagram provides a high-level inventory of actions among the actors on the battlefield during the mission execution phase of the mission. We focused on actions that interface with the wireless network. Each of these actions (later presented in the form of UML Use Case diagrams) cause the actors to generate network traffic.

In Chapters 4 and 5, we elaborate on this discussion through the development of a domain model and simulation model, respectively. In Chapter 4, we present the analysis of the fighting force, in terms of roles, formations, patterns of movement, and the artifacts exchanged during the movement and transition between patterns. In Chapter 5, we present the network traffic model derived according to the analysis of each Use Case in the Sequence diagrams that will be discussed in Chapter 4. The Sequence diagrams we present later in this thesis will elaborate the fighting force’s actions at the *squad* level; however, our intention in adopting the methodology approach in this thesis is to be able to exploit the natural scalability from squad to platoon level and higher, supporting the incorporation of these larger units in the model, without a change in the sequence of interactions defined. For example, the squad leader can be replaced with platoon leader and the team leaders A and B can be replaced by the first, second and third squad leaders. Correspondingly, teams A and B can be replaced by squads one, two, and three.

CHAPTER 3

DISCUSSION OF THE WIRELESS NETWORKING ENVIRONMENT

3.1 The Promise of Wireless Networking

For the delivery of wireless infrastructure on which to build platforms, applications and services, there are competing approaches (and their business models) that seek dominance in the industry--namely, 3G wireless networking schemes built around existing circuit-switched cellular voice networks, and wireless local-area networking (WLAN) schemes built upon the existing packet-switched infrastructure of the Internet (using the IP protocol).

Each scheme has its strengths and weaknesses [ISR02] [WMT01], with both offering to deliver the benefits accrued in their respective market spaces. For the voice-based wireless cellular communications vendors and service providers in the 2G-3G-market space, the goal is to provide data services and technology to allow Internet applications and content to be delivered on wireless cellular phones. This is already happening in certain markets, and the major phone vendors are working with service providers to extend the service offerings to take advantage of new technology, such as embedding Java to run web-enabled applications [EAB02].

For the data packet-based, web-enabled, content-rich applications and infrastructure of the WLAN-extended Internet space, the goal is to extend the set of content-rich applications to wireless PDAs and other devices, and to also incorporate voice through technologies such as Voice over IP (VoIP) and speech processing/natural language recognition [WMT01]. This civilian objective is similar to that of the military—namely, the formation of ad-hoc networks and exchange voice, data and multimedia information [KOL00] and to provide this environment with high levels of bandwidth and information assurance throughout the system [ARM08].

WLAN holds great promise for the Army as a means to extend the Global Information Grid (GIG) to individual infantry soldier in the field. The GIG is a worldwide communications network providing seamless, end-to-end connectivity for all warfighting, national security, and support users. [JRO01] To extend the GIG to the soldier, the Department of the Army has developed the Land Warrior (LW) system as a wearable system of sensors, a radio system, and a computer system designed to increase the soldier's situational awareness [HSM02]. The LW communications system is built on the IEEE 802.11protocol suite [HSM02] and enables the soldier to transmit and receive voice and data information critical to the mission [ORD01]. Every soldier in the unit is considered a node within the network, thus providing network connectivity at the team, squad, and platoon levels [HSM02].

3.2 Technology Objectives for a Wireless Infrastructure

In this section, we discuss our interpretation of the Fighting Force need in terms of several important architectural characteristics of robust systems—namely, reliability, availability, security and integrity, flexibility and maintainability. There are many architectural dimensions over which we might consider the Army's request, but these will illustrate the parameters that we will be considering as we embark deeper into our research effort. Finally, it goes without saying that these factors are to be considered in the context of managing the tradeoff between performance optimization and resource conservation in the wireless network.

3.2.1 Reliability

In considering this dimension, we are looking beyond MTBF and MTTR metrics, as such standards are well entrenched in production for military applications. Rather, we are interested in what types of system structure and behavior can be adopted to maximize reliability by minimizing exposed points of failure that might be susceptible to attack. In most networking systems, there is a mix of hardware and software components. There is a network processor or standard CPU, firmware or embedded software, among other components.

The possibility of integrating the wireless protocol stack into a hardware package without requiring software components creates a package with higher reliability possibilities (the exact estimates for such packaging for reconfigurable computing devices is published based on the specific vendors PLD and FPGA device components, by the vendors such as Xilinx and Actel).

3.2.2 Availability

The availability of the wireless network can be discussed in terms of the availability of a given node in the network, or the availability of transmission capacity over which variable loads can be carried (i.e., the “network” itself). In ad-hoc, multi-hop networks, the capacity of the wireless medium can be affected by a number of factors: (1) number of nodes, and their placement relative to one another (which affects channel congestion), (2) the frequency bands, their bandwidths, and how many channels are available for use (affecting the protocols that can be used for interoperability), (3) the power budget required at the node to push traffic onto the medium, and (4) the distribution of traffic on the available wireless media.

Availability can likely be maximized through some of the following ideas: (1) allowing individual nodes to be multi-functional, and allowing them to change function—as from an 802.11 station to an access point—in the face of changing environmental conditions such as traffic load, (2) allowing groups of stations to agree to switch to a different frequency band in the face of congestion or perceived threat, and supporting multiple PHY radio implementations in the same station, (3) allowing stations to unilaterally, or by agreement among groups of neighboring stations, to either cut or boost power to compensate for some environmental change—as might be the case if jamming, congestion or other disturbance were affecting the stability, throughput or QoS of the channel, and (4) again, allowing stations to alter their behavior—unilaterally or by agreement—during different periods of predicted or unanticipated load.

Another meaning of availability deals with the availability of “assets” of the network to authorized parties [PFL97] through some authentication and access control mechanisms—where these same resources are not available to a party that should not have access (a “denial of service”), and where the system knows the difference. This is tightly woven with security.

In addition, availability could apply to insuring that the network nodes share and use resources equitably, and that the resources are tolerant of faults arising from changes in the environment. Here, the critical resource is the wireless medium itself, although other critical resources could include available buffer memory with which to store and forward frames for a node that is in stand-by mode. In an adaptive, self-organizing network, we might expect nodes to collaborate to insure that availability in such scenarios is preserved.

3.2.3 Security and Integrity

The security and integrity of the network can be discussed in terms of physical security of the wireless devices themselves. It can also be discussed in terms of the security of the data transmitted over the medium. The integrity of the node implies that it is operating consistently with its programming and is not able to be tampered with, and that its assets can be modified only by those authorized to do so. Integrity of the data transmission is generally verified by the check sequence sent with frame data, according to CRC or other such scheme.

The security of the device implies that the packaging itself is tamper-resistant. The packaging might be fitted with sensors that could detect certain types of tampering, thus signaling the device's internal logic to take some specific action—such as alerting other nodes to the fact that it "believes" it is being tampered with. The security of the nodes themselves, the network, the data transmitted, and resources that are accessible through the network connection are all governed by basic notions of security, applied with the intention of providing controls to reduce vulnerability of the node, the data stream, the network and resources accessed through it.

There are a number of threats to the security, availability and integrity of the wireless network [PFL97]:

(1) Interruption of service - where a node, transmission, resource or other asset is unavailable, lost or unusable. This can be affected by jamming, introducing errors into frames, or attacks on stations, access points, or other resources in an attempt to compromise the integrity of the resource or system.

(2) Interception of an asset – where an unauthorized entity has gained access to the asset—a frame, resources of a node, etc.

(3) Modification of an asset – where an unauthorized party modifies an asset in an attempt to thwart or compromise the integrity of the system.

(4) Fabrication of counterfeit assets – where an unauthorized party creates assets—frames, signatures of actions, etc.—that are false, with the intent to mislead or misdirect.

Security is managed in a network environment through a number of different means: (1) identity-based authentication of messages, senders and receivers, and any other identifiable resources involved in transactions, (2) identity-based, or role-based, access control, (3) data encryption, in both the transmission medium and inside the stations themselves (to retain confidentiality in the face of direct attack on the device and/or its packaging), (4) integrity and non-repudiation through checksum schemes and signatures applied to the data stream and to the internal contents of the nodes (such as the logic and memory contents), so as to have high confidence that the recipient node is receiving something from the claimed sender, and (5) audit trails tracking a history of actions for *a posteriori* access checks.

All of these schemes are applied at layers above the Link Layer of a networking protocol stack. At issue is which measures can be applied at this level so as to improve security, integrity and availability of the network assets [KUR03]. Today, the 802.11b protocol has checksums and encryption applied to the frame data stream. However, we believe it is possible to provide additional capabilities to make the MAC more aware of potential threats, able to assess likelihood of use for threats, and apply overlapping controls to limit exposure. Adding new capabilities and sensitivities to possible threat scenarios will be considerable part of our analysis and development of adaptive behaviors in ASOWN.

3.2.4 Flexibility and Maintainability

In considering flexibility, we are interested in the ability of the network and nodes to adapt to changes in the environment. Most systems handle change through direct modification by human systems developers. Protocol changes, software upgrades, and the like are part of this process. It is our intention to explore in what ways we can make the network more resilient and flexible to planned change—such as by facilitating secure means to update the configuration to the logic circuits as planned changes are instituted. In addition, we are interested in how to make individual nodes, and the wireless network as a whole, more flexible in the face of environmental change.

3.3 Overview of Adaptive, Self-Regulating Systems

Adaptation is defined as any process operating on some structure where the structure is modified over time to improve its performance (of "fitness") in its environment. A structure's adaptations to its environment are "persistent properties" of the structure created as a result of its behaviors, where these behaviors act as a set of "structural modifiers" or "operators", applied according to some pattern (or "plan").

This notion of adaptation can be applied to systems, populations, individuals, or components thereof. The most commonly studied systems are insect colonies [GOR99] [JOH01], cellular organic systems [RUM86] [JOH01], or other systems involving large populations [AXE97]. These systems have a set of the following salient properties [HOL92] [AXE97] [JOH01]:

(1) They are very decentralized, generally without any "controller" function coordinating activities of all the population members (regardless of a role designated as "Queen");

(2) They are capable of using small adaptations (through simple learning maneuvers) in individual behavior to modify the collective behavior on a global scale in response to changes in their surroundings as perceived at the individual's level of operation in the population;

(3) In certain populations, individual members can change their behavior or function in the presence of indicators or signals triggering them to do so, based on what their neighbors are doing;

(4) In self-organized systems, the structural patterns exist at multiple levels of system abstraction, where the actions of a higher layer constitute the environmental changes of the lower layer, whereas behavioral changes at a lower layer percolate and propagate through the population of members at that layer to affect change in the layer above it; and,

(5) In certain systems, introducing random mutations in addition to environmental changes can cause new behaviors that may or may not be beneficial to the population; however, the reproductive effectiveness of the individual (or the propagation of the learned behavior through the population) based on the fitness imparted by its behaviors determines whether the trait will manifest itself in the wider population. It is not clear whether such assumptions of a “genetic” model are appropriate for considering adaptation and self-organization in wireless networks. This remains to be seen.

In the following sections, we summarize some of the properties of adaptive, self-organizing systems, how they may relate to Link Layer MAC functionality in a wireless network (using the 802.11 in particular), and how we might exploit these characteristics in the ASOWN extension of the 802.11 MAC layer to exhibit self-organization as a scheme to address security, capacity/resource and performance management limitations in using ad-hoc wireless networks with large populations of nodes.

3.4 An Agent-based Model for Adaptive Node Behavior

In the ASOWN research project, we are interested in creating a large population of individually acting wireless agents whose collective behavior improves the overall fitness of the wireless network landscape. This improvement in fitness may be defined in terms of performance, resource conservation and security threat avoidance. In consideration of this objective, it helps to examine the relevant characteristics of an agent-based model for system organization.

We can graphically explain the research landscape for ASOWN in terms of a software agent model, as depicted in Figure 16, depicting a general view of architecture—the structure and organization of functional components used to exhibit behavior in an environment—using the model of evolutionary biology as a guide [DEN95]. Evolutionary models have long been the basis for much agent research [MAE95].

A software agent can be modeled on a biological organism, with salient characteristics extrapolated and mapped onto the domain of a software program. As with most higher level organisms, agents have the following: (1) internal needs and desires, (2) practiced means to satisfy those needs and desires using available resources, (3) means to sense what's going on in their surroundings, (4) means to evaluate perceptions for purposes of selecting appropriate response, (5) means to affect change in their environment relative to their needs and desires or relative to perceived environmental stimuli, and (6) means to evaluate the results of that change in order to move them closer to achieving a specific *need or desire.*

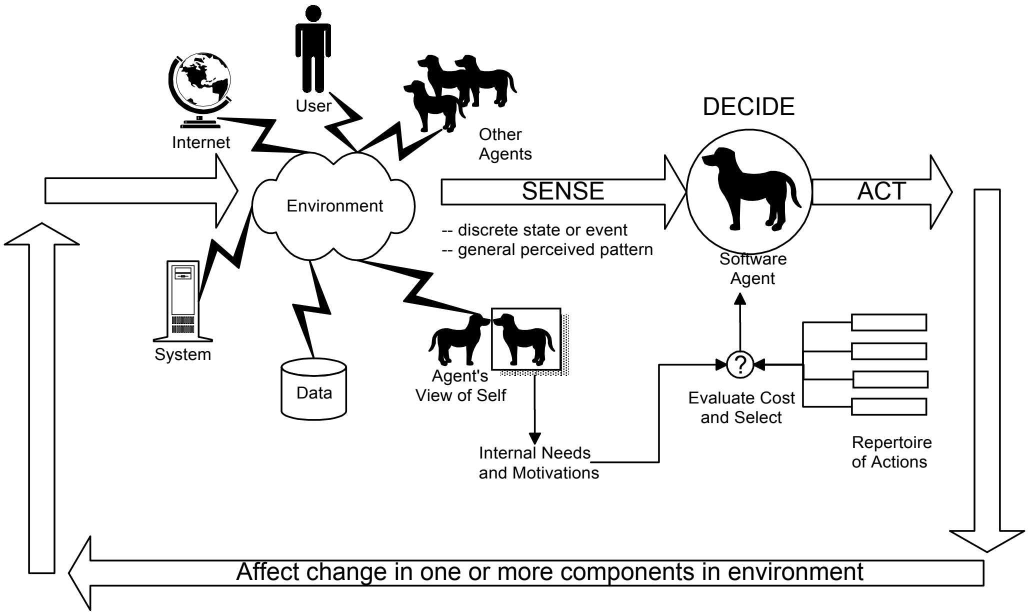

Internet
User
Other Agents
DECIDE
Environment
SENSE
ACT
-- discrete state or event
-- general perceived pattern
Software Agent
System
Data
Agent's View of Self
Evaluate Cost and Select
Internal Needs and Motivations
Repertoire of Actions
Affect change in one or more components in environment

Figure 16. General Architecture for Autonomous Software Agents.

In nature, biological differences between organisms center on "behavioral machinery" and "architecture" tradeoffs evolution has made for coping with the complexities of their respective environments, where behavior results from applying the machinery. Higher-level organisms have more varied and complex means for sensing, evaluating and responding to their environments than do simpler organisms. Mechanisms such as memory and feedback enhance the level of response by giving the organism a means to build on prior experience of interacting in the environment.

An internal representation of other organisms in its environment, and the means to communicate with others of its kind, provides opportunities for the organism to cooperate on satisfying innate needs (e.g., procreation to extend its genetic line). Richer biological constructs allow greater manifestation of "intelligent" behavior—such as through self-awareness, advanced learning strategies, advanced visualization techniques to "see" future outcomes (i.e., as a "plan") in its environment. Finally, it is often the case that organisms enter into symbiotic relationships with organisms of other kinds in order to achieve mutually beneficial objectives; strategies for cooperation and collaboration are important for survival and mastery of environment.

By abstracting these basic mechanisms in living organisms, we can discuss a simple and general architectural model for our agents. First, agents must have some means to sense their environment. In our context, this notion of environment includes: (1) data in the received MAC layer frames, internal MAC layer, PHY layer and other "data sources;" (2) explicit goals, tasks and actions and implicit intentions of the soldiers or the network as a whole; (3) events or information generated by the Army Battle Command

System (ABCS) applications and lower-level software modules in the multi-tier system on which the agents execute; (4) events or information from resources on the Internet; (5) explicitly-represented information about the agent's own goals and plans; and (6) explicitly-represented information about the goals, tasks and intentions of other agents in the environment.

The sensation of information from the environment can come in several forms: (1) discrete "events" from the environment (i.e., something that the agent is capable of sensing has been detected, such as the presence of a specific frame type being detected in the network traffic); (2) discrete "states" of artifacts in the environment; and, (3) general patterns perceived as a result of historical and ongoing interaction with the environment.

This last case presents an interesting scenario, namely, a station realizing that a particular sending station is the originator of an exceptionally high number of frames with integrity errors, as indicated by failure of the MAC's FCS decoder to generate a consistent CRC value for received frames. If such a condition were perceptible by the MAC layer, the MAC might develop a hypothesis that some entity is interfering with the integrity of the data transmission of this sending station (with some likelihood value). Alternately, it might conclude that the sending station's identity might be questioned suspect (with another likelihood value).

Once stimuli from the environment have been detected, identified and classified, the agent must decide what to do in response to each stimulus (either considered separately or collectively). In this regard, the agent might need to do one of the following: (1) react to externally-generated changes in its environment; (2) evaluate changes created by its own actions as a result of prior actions taken to further its own internal goals (what to do) and plans (how it is to do it); or, (3) evaluate its own performance on tasks in pursuit of its goals to see how to adapt its own behavior to improve performance.

In all cases, the agent must do the following: (1) look at its own state, comparing internal notions of goals and plans against perceived, identified and classified changes in its environment; (2) identify actions from its plans—or generate new unique plans--along with the "cost" or "utility" associated with taking each action, as part of formulating a set of "candidate" actions to consider in response to the environmental changes; (3) select one or more actions from the repertoire of possible responses, as a result of maximizing utility or minimizing cost, and schedule them for execution; and (4) execute the scheduled actions on the environment in the appropriate order.

This *sense-decide-act* process repeats, where the agent's affecting its environment is assessed as a result of sensing (or not sensing) change in that environment. When the agent is responding to external stimuli, it is engaging in "regulating" behavior—a primitive (and, still, most prevalent) form of agent behavior available in agents today [BRU91].

When the agent is executing actions as part of tasks it executes in pursuit of its own goals, it is engaged in either "reasoning" behavior [FRA96] (inferring new information about the environment based on matching what it perceives in the environment against what it already knows) or "planning" behavior [ETZ94] (creating or changing the sequencing of tasks to achieve its goals in response to new information), or both. When an agent has the ability to communicate and work together with other agents, it is engaged in "collaborative" behavior [GEN94]. Finally, when an agent is modifying its own strategies for interacting with its environment—in terms of its ability to respond, reason or plan effectively--for the expressed purpose of improving performance, it is engaged in "adaptive" behavior [MAE94] (i.e., it is capable of "learning" and applying new knowledge).

3.5 Behavior of the IEEE 802.11b MAC Protocol

An important point of our research is the identification of appropriate and suitable models and architectures for adaptive self-organizing behavior in the wireless network, then to make changes to the 802.11 MAC with the intent of modeling a modified version of the MAC protocol in *ns-2* with which to perform experiments. Other research has focused on adding new layers to the protocol stack [LEE99] [HSM02] so as not to affect the existing protocol and architecture of wireless networks. We believe that the types of changes in station and network behavior are only possible by modifying the core MAC protocol and architecture itself.

There are several reasons for this position: (1) there are "purist" reasons, in that a separate layer having access to internal state of the MAC violates the principles of separation between layers in a protocol stack, (2) there are security reasons for integrating the modifications directly into the MAC, as having such knowledge outside the layer would call into question the "boundary of trust" that exists between the MAC and entities outside it, and (3) there are performance reasons, in that the computational overhead of providing adaptive, self-organizing behaviors will require more computational power, more memory and other computing resources. These resources can be made available through a reconfigurable VLSI computing architecture, but the partitioning of functionality, along with the overhead to maintain security of its internal structure, adds a performance penalty.

We will begin this discussion with an explanation of two methods the MAC layer uses to provide access to the medium. Next, we will explore how the principles of adaptive, self-organizing systems can be applied to wireless network. Finally, we will speculate on the types of modifications we envision should be supported through use of the simulation model created in this research to make to the 802.11 MAC layer more adaptive and self-regulating in the battlefield sphere.

The MAC uses two different coordination functions to regulate access the physical medium: the distributed coordination function (DCF) and the point coordination function (PCF). The DCF is the standard access mechanism for the wireless nodes and can be used in ad-hoc or infrastructure networks. This DCF forms the basis for the PCF, which is only usable on infrastructure networks and is usually implemented in an access point (AP) of a basis service set (BSS). The PCF uses an operation similar to polling to determine which station has the right to transmit [ANS99]. Figure 17 shows the “stacking” relationship between the DCF and PCF.

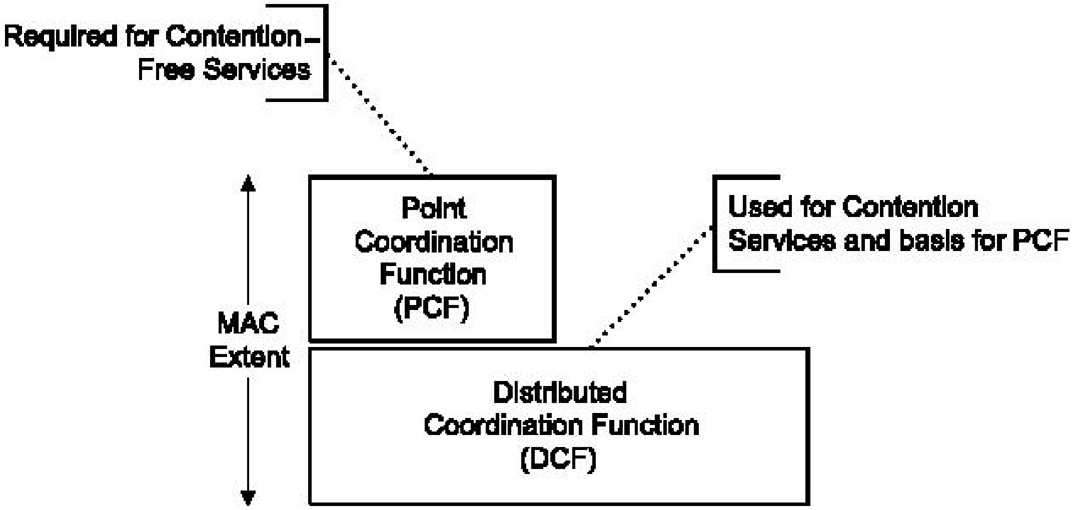


Figure 17.

DCF mode: The DCF provides an Ethernet-like contention-based service by using the carrier sense multiple-access with collision avoidance (CSMA/CA) protocol with a random backoff time after a busy medium period [GAS02][ANS99]. The CSMA/CA protocol senses the network through both physical and virtual means to determine if the network is busy. If the network is busy, then the node waits until the network becomes idle, plus a random backoff time. The random backoff time reduces the probability of a collision between multiple stations attempting to access the same medium as soon as it becomes idle [ANS99]. The MAC also uses a variety of interframe space times, inserted before the backoff time, to allow for higher priority traffic [GAS02].

Physical carrier sensing protocols depend upon the medium and modulation used, therefore it is implemented at the physical layer. However, virtual carrier sensing occurs at the MAC layer using the network allocation vector (NAV), a timer indicating the amount of time a station will reserve the medium [GAS02]. Each frame sent by a station includes a Duration/ID field, which sets the time required for transmission of the data frame and the resulting ACK. Every other node receiving this frame sets its NAV to be at least as long as the Duration/ID field. The stations then count down from the NAV to 0. An NAV of 0 indicates that the medium is idle while a non-zero NAV indicates a busy medium. Using the NAV ensures that atomic operations are not interrupted.

After the NAV, a station must wait for the period of time determined by the inter-frame spacing. The 802.11b protocol uses four different types of inter-frame spacing to create different priority levels between different types of traffic. The three inter-frame spacing times shown in Figure 18 are used to determine medium access. The fourth is used only when there is an error in transmission and is not important to our research at this time. Normally, atomic operations start as low priority transmissions, so they must wait until after the DIFS to transmit. However, subsequent transmissions within the same atomic operation are treated as high priority transmission and need only wait until after the SIFS [GAS02].

Short Interframe Space (SIFS)	Used for highest-priority frames, such as RTS/CTS and positive acknowledgments. The highest priority transmissions can begin after the SIFS has elapsed.
PCF Interframe Space (PIFS)	Used by the PCF to allow stations to transmit during a contention-free period. Stations with data to transmit during the contention-free period can transmit after the PIFS has elapsed.
DCF Interframe Space (DIFS)	The minimum medium idle time for contention-based service. All stations may have immediate access to the medium after the DIFS has elapsed.

Table 3. Spacing delay between frames (after [GAS02]).

After the NAV and DIFS have elapsed, nodes may transmit based upon their randomly selected backoff time. The time after the NAV and DIFS have elapsed is called the backoff window, which is divided into numbered slots. Each node chooses a random number from the backoff window and is allowed to begin transmitting in its slot as long as no other node with a lower number transmits a frame. If a transmission fails, then the backoff window grows.

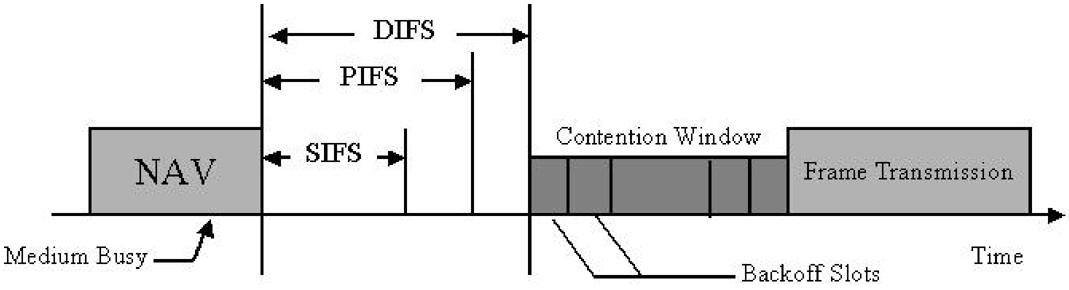


Figure 18. Frame transmission timing constraints.

Backoff windows sizes are always 1 less than a power of 2 (e.g. 31, 63, 127, etc). Therefore, upon transmission failure, the backoff window increases to the next power of two. If the transmission is successful, then the backoff window shrinks to its minimum size. The physical layer limits the maximum backoff window size. Figure 19 shows the timing relationships between the NAV value, the inter-frame spacing delays, and the backoff window.

PCF mode: As shown in earlier figures, PCF (point coordination facility) shares many of the same features as the DCF mode of operation. However, one node, designated as the access point (AP), uses the PCF to provide contention-free service periods, during which other stations may only transmit data at times designated by the AP. The AP enforces “fair” access to the medium through a scheme resembling token-based medium access control schemes; however, the station polling by AP takes the place of passing the token. The AP uses the “toolset” from the DCF (NAV, inter-frame spacing, etc), but in a slightly different pattern to lockout access to the medium by other nodes [GAS02].

When the PCF is being used, the AP divides time into contention-free periods (CFP) and contention periods (CP). During the CFP, access to medium is determined by the PCF. At other times outsides of the CFP, medium access reverts to the normal DCF methods until the next CFP. At the beginning of the CFP, the AP broadcasts a beacon frame containing the maximum duration of the CFP, CFPMaxDuration. All other stations set their NAV to the CFPMaxDuration time. This effectively locks out these stations from transmitting. Additionally, all transmissions during the CFP are separated by the SIFS and PIFS. Since these inter-frame spaces are shorter than the DIFS, this also reserves the medium during the CFP.

After the beacon, the AP begins polling associated stations on its polling list to determine if those stations have any data to transmit. If station one has data, it transmits to the AP after the beacon and the SIFS. The AP then sends an acknowledgement to station one and polls station two. If station two does not have any data to send, the AP will send a poll to station three after the PIFS. Polling continues until the actual end of the CFP or the AP sends a CF-End frame. In order to maximize the throughput during the CFP, different frames will be piggybacked onto each other. For example, in the above scenario, after the AP receives data from station 1, it can send one frame containing an acknowledgement for station one and a poll for station two. Figure 20 shows frame sequences during a CFP and CP.

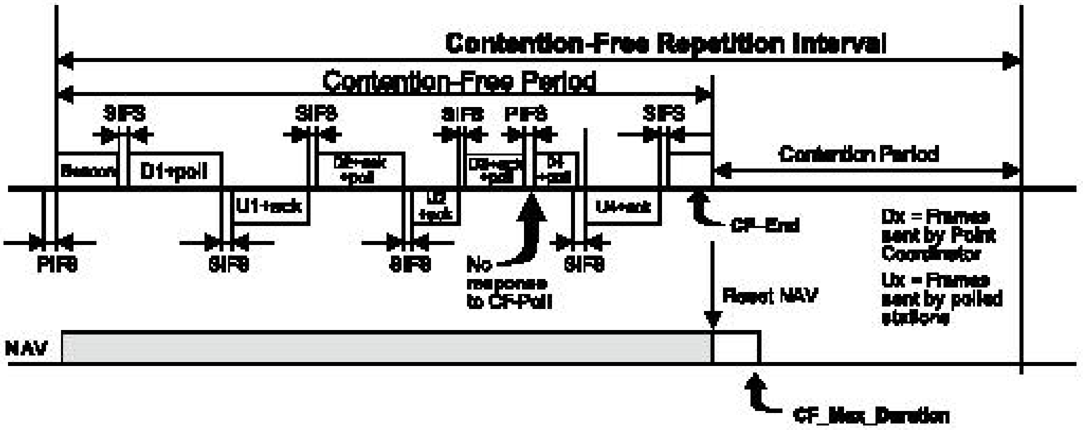


Figure 19, Inter-frame delays (after [ANS99]).

3.6 Adaptive, Self-Organizing Systems

There are five relevant principles of adaptive, self-organizing systems we will use to conjecture some of the possible avenues for creating an adaptive, self-organizing wireless architecture. The five principles are the system lacks a centralized controller, system behavior is the sum of individual behaviors, individuals can change their behavior dynamically, adaptation and learning occur on different levels simultaneously, and individuals may choose to "defect" from social interaction norms in order to achieve certain objectives.

3.6.1 Centralized control

The actions of individuals in an adaptive, self-organizing system are scripted by simple hardwired plans, whose execution is triggered by a set of rules subject to perception of certain randomly occurring environmental cues [JOH01]. Individuals track a number of specific types of sensory inputs, levels of intensity of inputs, and count the frequency of certain events on those inputs, and correlate their behavior to the inputs.

Similarly, wireless network protocols, such as 802.11, have no central controller, but have simple hardwired functions embodied in the protocol definition. However, unlike individuals in a colony population, wireless stations are only adapt to changes in their environment to that degree based on the foresight of its designers.

As nodes “randomly” move into and out of ad-hoc configurations, they listen in on all frame traffic to which they can sense so as to ascertain whether it is frame traffic being sent to them. They do minimal processing on a frame that is not identified as theirs (e.g., updating NAV Register to reset the wait time before attempting to access the medium). If the nodes were changed to do more than the minimal processing on frames it would otherwise throw away, can the nodes use any data gathered from those frames to respond more flexibly to the random changes in their environment.

The major areas where this might occur are bandwidth, traffic congestion management, security threat detection and internal resource optimization. If the MAC monitored inputs, such as frame traffic, from its neighbors, and did more than simply throw it away once it extracted the NAV value, it might, like an ant, be sensitive to gradients, such as signal strength. What if a node kept track of occurrences and frequency of occurrences, for events in its iBSS/BSS/ESS? Could individual MAC nodes detect meaningful patterns in its environment? Could it affect its environment in an adaptive way, by using this new sensory information, to optimize its own performance or detect threats, from its limited point of view?

3.6.2 Sum of Individual Behaviors

An adaptive, self-regulating system is also characterized by the fact that the system’s behavior at a macro level is created by the summation of the individual behaviors of its individual members [JOH01]. There is systematic complexity, yet order is built out of local interactions between individuals [HOL92]. In a WLAN, the functioning of the network (iBSS, BSS or ESS configuration) depends on the individual node's conformance to the IEEE 802.11 protocol. WLAN nodes operate in an environment of limited individual resource capacity such as bandwidth of the medium, energy availability (battery life), amount of buffer space, etc. Excessive congestion at one node may cause feed-forward effects on bandwidth management which effects quality of service for the entire network. For example, excessive waiting traffic could adversely impact frame delivery, causing excessive retries, further clogging contention for the medium, affecting network quality of service. The primary issue here is what happens when a very large population of nodes is present, thus taxing the available medium bandwidth? What if terrain conditions, or threats, cause rapid, dramatic change in its environment? What happens when there is rapid shifting of the population due to high mobility, and high-speed mobility? What is the effect of individual node movement on the capability, capacity, reliability, security and maintainability of the entire network?

3.6.3 Changing Individual Behavior

In an adaptive, self-regulating system, individual members of the population can change their behavior dynamically, to take on different roles, based on local changes in their environments. It has been documented that forager ants—on detecting certain concentration of other foragers, will change behavior to that of another role in the colony. Over time, roles played by an individual may change [GOR99]. In 802.11, the function of nodes (as stations, access points or bridges) is hardwired. The protocol defines a means to change to basic mode of operation (e.g., between DCF and PCF modes) to alleviate contention. A station operating in the DCF may request that the AP set up PCF so that each station must request, and send traffic, through the AP—in effect, setting it up as a "controller". This allows different traffic patterns to be established, so the medium becomes an arbitrated resource. But, we presume that, with a large number of stations in a BSS, this will cause severe traffic backlog, resulting in excessive timeouts, and dropped higher-layer packets. What if, based on monitoring traffic patterns in the frames received, and its own objectives, a node can change from a station into an AP? What if the conditions under which such a transition proved beneficial could be learned by the node and passed onto other nodes in the network? What if a node, by observing frame integrity and encryption checks (FCS and WEP frame fields), could detect a pattern of security threat or breach, and could set up tests to confirm its hypotheses, then communicate its findings to other nodes? What if it could devise new strategies to thwart attacks, jamming and other threats, and thus passing these insights to neighboring nodes? Also, how might this trait affect multi-hop performance in an ad-hoc network?

3.6.4 Recurring Patterns

The nature of adaptive, self-organizing systems is there are recurring patterns at different levels of organization, and that the adaptation and learning occurs on different levels simultaneously, through the modification of some structures at each level by operators specific to that level [HOL92]. In networking there is a hierarchy of abstractions, involving the network as a whole (such as the Internet itself), which consists of other networks linked by routers. In the wireless network component, the routers and nodes are stations. It is conceivable that we can think of the internal architecture of the MAC as a set of nodes that communicate among one another via another protocol (as it typical of computer architecture and VLSI devices). Can this principle of layered abstractions, with adaptation occurring at micro and macro levels be harnessed to improve the fitness of the network at these different levels of abstraction? Can effects of local behavior changes in one level promote better performance in the wireless environment, or of the larger network as whole?

3.6.5 Controlling Interactions

Self-regulating systems use behavioral "norms" to control interactions among agents, where, in certain social situations, individuals may choose to "defect" from these norms in order to achieve certain objectives [AXE97]. Considering norms in terms of precisely defined protocols, they can be *arbitrated*, *adjudicated* or *self-enforcing* [PFL97]. IEEE 802.11 MAC behavior constitutes a protocol, or an orderly sequence of steps that all parties must follow to accomplish a task, where the regulated behavior is to the benefit of all [PFL97]. The MAC protocol is fixed around certain agreed-upon conventions, such as CSMA/CA, to guarantee that the wireless medium is shared fairly. In addition, there is implicit prioritization of traffic based on using inter-frame spacing [GAS02].

What if, under certain circumstances, a given node could decide to "defect" from observing CSMA and CA so as to maximize its own access to the medium? What if the node's neighbors (i.e., other stations in the iBSS or BSS) were able to punish it under certain conditions? What if a station could assess its local conditions, for reasons of threat/security or resource management, and decide to "coerce" the protocol to satisfy some pressing objective?

3.6.6 Relevant work in adaptive systems

Other work in this area focuses on using adaptive agents to optimize search problems. The ACO (ant colony optimization) [BON00] is based on optimization to find shortest-path solutions for a broad class of problems. A derivative of this technique, referred to as ACR (ant colony routing) [HEU98] has even been applied to packet routing and load balancing in networks. The difference between the research we are conducting and the prior work cited above is that we don't believe our problem is fitted to a shortest path optimization problem. We are basically dealing with a problem where there are a number of small, interacting sub-problems involving specialized adaptations of a wireless node to its changing environment. Basically, our agent (i.e., a wireless station) is actually a collection of agents that cooperate on tasks pertaining to threat detection ("patroller" agent), power and resource management ("foraging" agent), congestion management ("maintenance" agent), exception handling ("midden" agent) and MAC function management ("nest" agent). This taxonomy, although somewhat arbitrary at this point, follows the classification of ant roles observed in the field [GOR99].

Furthermore, just as there are multiple agents operating within the confines of a specific MAC unit, there are corresponding agents in each MAC unit. Thus, in a given high-density multi-layered network topology, there could be thousands of such agents interacting, collaborating, cooperating and competing with one another to achieve specific objectives associated with their own functioning—as dictated by the environmental landscape. We are assuming that results from earlier work on this type of agent architecture, from the healthcare domain for medical protocol compliance [DAA98] [DAV98] [DHB98] would be applicable in creating the overall agent-oriented architecture in the wireless LAN environment. As such, our research problem appears to be nothing like the shortest path optimization problem of the ACO technique. However, there are likely to be formulations for cooperative learning, planning and diagnosis problem-solving activities to be performed by the agents—learning being fundamental to the adaptive character of the wireless network at both the individual node and for the network as a whole.

3.6.7 The MAC as an Agent

For the purposes of this research, we want to focus on using the MAC layer of the node as an agent, whose purpose is to gather statistics relating to network quality and, if those statistics go above or below a certain threshold, change the MAC layer operation from DCF to PCF in order to improve network quality. In using the MAC layer as a collection of agent-based behaviors, we posit the following questions:

One question of interest is whether the MAC layer can adapt to congestion conditions, for instance, by initiating a change within a given node itself from DCF to PCF and back, without having a dedicated node playing the role of an Access Point? Theoretically, yes, because PCF mode is initialized by an Access Point sending a Beacon Frame [ANS99] that locks out other stations from sending traffic onto the WLAN. During this time, the AP can poll other stations in the BSS for traffic. Other stations entering the BSS during the NAV time set by the beacon are locked out from sending traffic since the new station has to wait until after the DIFS to transmit and the AP transmits after the SIFS and PIFS. Once the CFP expires, medium access reverts to DCF mode, and the node that assumed the AP role can relinquish this responsibility, or continue to monitor its environment in the AP role.

A related question concerns what would constitute an appropriate period of time for the NAV value set by the beacon frame? One possible answer is a standard range of incrementally longer times that the AP can cycle through until the network problem is resolved. It should be feasible to explore answers to this question through using the model being developed as part of this research as a basis for experimentation.

A second, broader question of interest involves what local information can the node sense in order to make determinations about the state of the network? We must define "local information." For the MAC layer, there are actually two kinds of local information: (1) statistics relative to the node, and (2) statistics gleaned from received packets (both packet destined for and not destined for the node). Information relative to the node includes packets received, packets forwarded, packets sent, packets dropped, ACK's received or not received, and queue size. In addition, the node can gather information such as the source node, destination node, packet's time to live, and byte size from each packet it receives.

Third, given that it is possible to modify the MAC layer functionality to gather statistics on network traffic and loading patterns, what do they tell us about the state of the network? Some QoS requirements like minimum acceptable rate and max delay are difficult for a node to detect except in terms of packets sent to its neighbors. For packets sent over multiple hops, it would be difficult for any one node between the source and destination to determine the global throughput or delay for that link. The source node may be able to determine the throughput on the entire link by dividing the number of data bytes sent by the time required for the entire atomic transaction. However, the problem with the link may be at one of the intermediate hops for which there is nothing the source node can do (i.e. no change the source can make which will affect link quality at another hop).

If we were to use the number of dropped packets as a metric, how does it relate to network quality? We have to look at why a packet is dropped. A packet gets dropped when the queue resource associated with a node is full. Now, what causes the queue to get too full? Is it a sudden increase in the amount of traffic on the network, or it is due to a "missing" node. Is the increase in traffic caused by real traffic or an increase in traffic going through a particular node while trying to route around the missing node?

It has been our objective to use our simulation model to provide a platform to explore answers to these questions on evolving the MAC protocol and architecture at the node level to add adaptive, self-organizing capabilities to the network as a whole. The subsequent chapters of this thesis show how we have defined the modeling activities and created the simulation model with the intent of using it as a basis for this type of exploratory activity.

Assessing the types of changes we might posit making to the MAC layer is based on collecting large amounts of simulation data, across a number of different model runs, then attempting to analyze the data to identify patterns in the behavior of the network. Battlefield scenarios can be diverse, as can be the conditions under which the network can be expected to operate. To be able to explore such changes easily, we need a model platform that is well-codified and easily modifiable, to support easy creation of model scenarios that can be subjected to simulation experiments.

Our approach has, thus, been to focus on creating a robust infrastructure for such exploratory modeling, where the necessary domain knowledge has been articulated into a set of extensible *tcl* scripts, which can be used as the basis for subsequent exploratory simulation. Describing the analysis and creation of the simulation model becomes the focus for the remainder of this thesis.

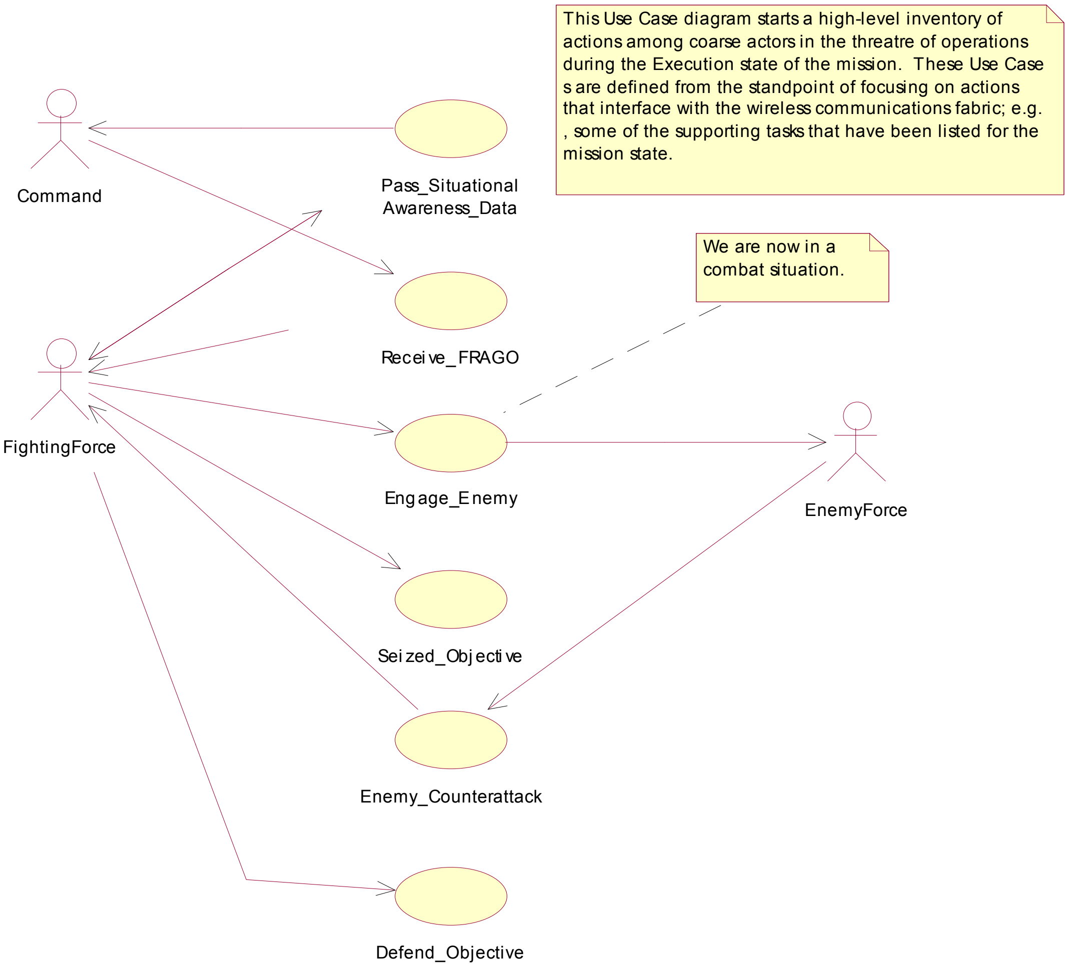
CHAPTER 4

PRESENTATION OF THE CONCEPTUAL MODEL OF THE DOMAIN

In this chapter, we present the conceptual model that has been created, based on the information about the fighting force—its formations and patterns of movement—as presented in Chapter 2. We use the Unified Modeling Language (UML) as the notation and method for capturing this conceptual model. This notation has been used in countless situations to create domain models; however, we have not seen its use directly applied to the analysis of a battlefield domain, nor have we seen it used as a "mediating" representation between the domain and the subsequent simulation models created from its conceptual models.

4.1 Modeling the Battlefield Domain with UML

The Unified Modeling Language (UML) has been developed with the objective of providing a multiple-view modeling notation and corresponding set of analysis methods for modeling complex domains. UML provides a means to capture the semantics of an arbitrary domain, represent its structural and behavioral semantics in a concise yet readable format, which can be both precise and correct (depending on the level of precision and accuracy desired in the modeling of an arbitrary domain of discourse). As such, UML has been used in thousands of application domains to describe the semantics of enterprises and the systems that operate in these enterprises.



This Use Case diagram starts a high-level inventory of actions among coarse actors in the threatre of operations during the Execution state of the mission. These Use Case s are defined from the standpoint of focusing on actions that interface with the wireless communications fabric; e.g. , some of the supporting tasks that have been listed for the mission state.

We are now in a combat situation.

Figure 20. Key actors and Use Cases for the battlefield domain.

Our interest in using UML in the context of this research is based on the following needs: (1) to effectively capture the underlying relationships between the constituent units of the fighting force organization in a general way, so as to not tie the modeling of a network to one specific unit size; (2) to capture the conceptual entities, their relationships, properties and key domain constraints, that can be used to guide the process of synthesizing a communications model for use as input to simulation; (3) to provide an means to explore the abstraction of model properties—which should be captured and which should be ignored—as a basis for starting the exploration of modeling for simulation, and, (4) to provide an economical means of depicting the underlying domain semantics to those outside of the military who would need to have this basic understanding in order to fully comprehend the exploration of the solution to the research problem in context of the problem statement itself.

4.2 Modeling the Organizational Units of the Fighting Force

We have presented the background of the fighting force and its purpose in Chapters 1 and 2. Therein, we discussed how the force carries out its mission objectives through coordination of activities in the movement of document artifacts among members of the force. We also discussed the how the force executes the mission through coordinated movement of soldiers on the battlefield as they transition between stages of the mission.

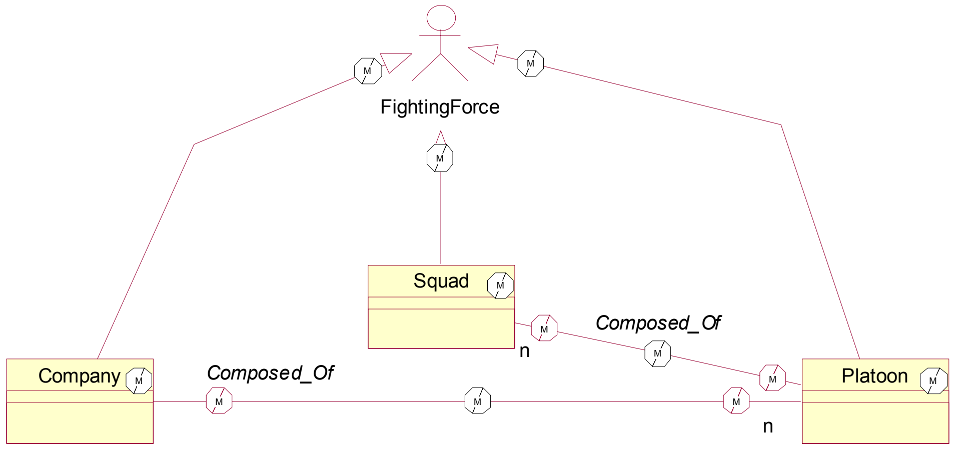


Figure 21. The Organizational unit hierarchy of the fighting force.

The fighting force “actor” can be decomposed into a set of constituent fighting force units, as discussed in Chapter 2. These units have a hierarchical relationship with one another, as indicated in the Class diagram in the figure above.

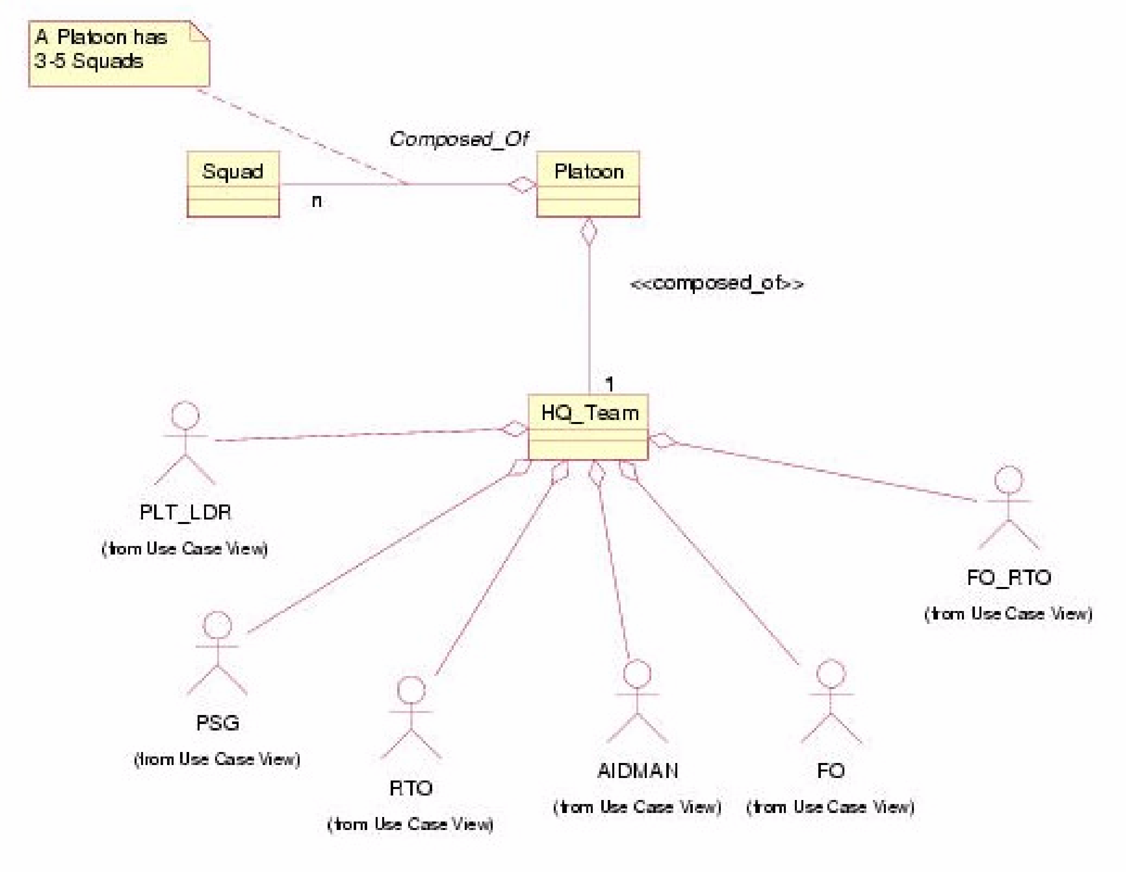


Figure 22. The platoon unit hierarchy in the fighting force.

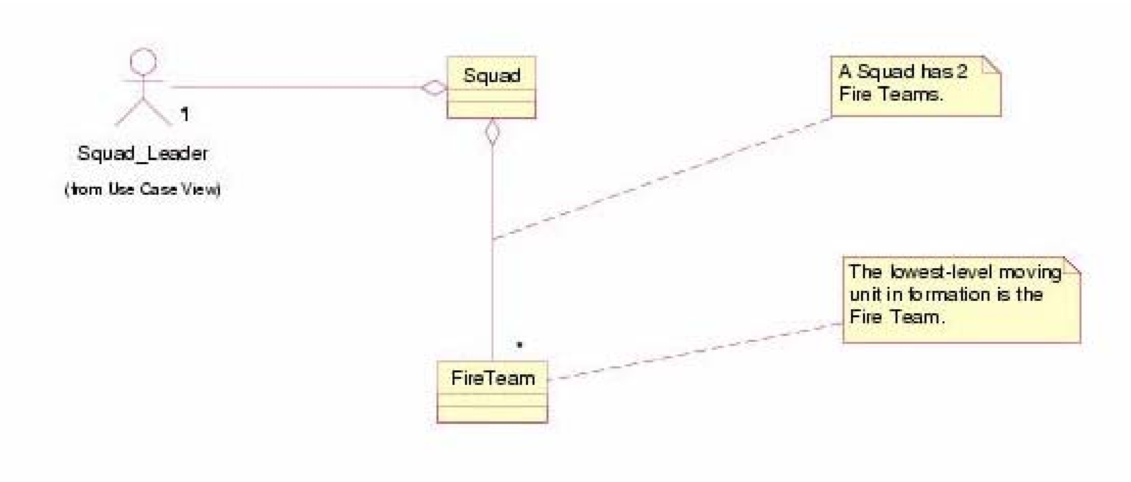

Squad_Leader
(from Use Case View)
1
Squad
A Squad has 2 Fire Teams.
The lowest-level moving unit in formation is the Fire Team.
*
FireTeam

Figure 23. The Squad unit hierarchy of the fighting force.

The UML Class diagrams depicting the decomposition of the various units into their components are shown in the accompanying figures 22, 23 and 24.

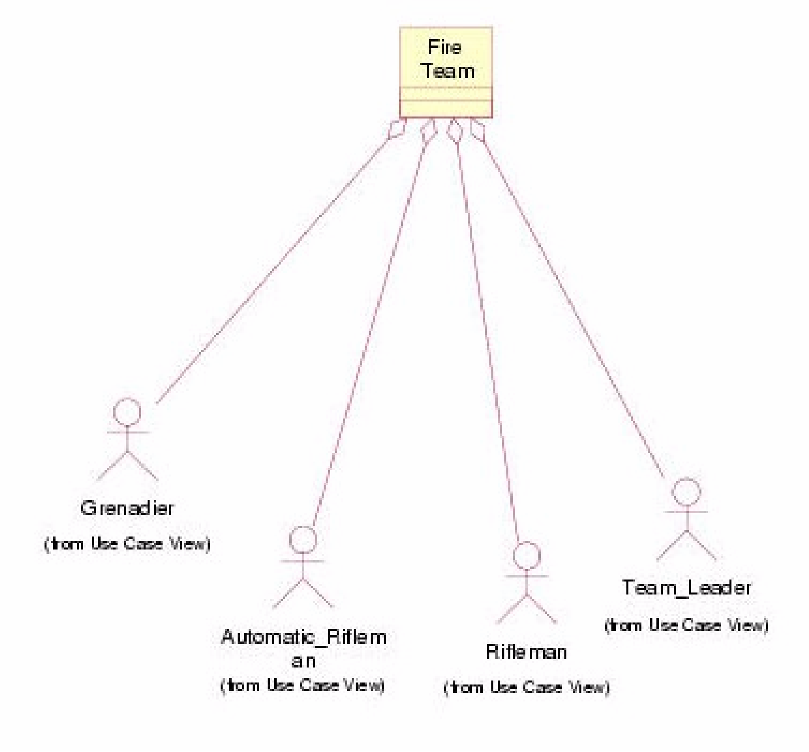


Figure 24. The Fire team unit hierarchy role-based decomposition.

Our gaining an operational understanding of the underlying fighting force domain was an essential part of the research work required to develop a simulation model accurately reflecting the battlefield communications network infrastructure. While we have shown how the Fighting Force actor decomposes into the Infantry Fire Team, the model also shows how the domain scales upward from the Fire Team. Just as infantry combat tactics are built around Fire Team movement and the relationship between Fire Teams, so is our model built on Fire Team movement and communications artifacts generated by soldiers within the Fire Team and passed through the network. Since we now understand how Fire Team movement scales upward to Squad and Platoon movement, we will be able to understand and model movement of larger fighting forces.

4.3 Modeling the Artifacts of the Fighting Force

The command and control infrastructure of a fighting force is maintained through a series of "artifacts" used to convey intelligence and command and control information to and from the command center and the organizational fighting units of the fighting force. These artifacts are mostly documents—many of which were described in Chapter 2. However, some of the information contained in these documents can be conveyed verbally.

What is important about the artifacts is that we have determined through our analysis that they represent—in an abstract form—the key transactions of interest in the battlefield enterprise domain. To state it another way, it is the transactions executed among the actors in the execution of the Use Case scenarios that represent the salient aspects of the domain. This is because, in considering the formulation of an appropriate network model on which to run simulations, we need to have some understanding of the network properties. To get this understanding, we need to be able to relate domain interactions with network parameters—such as the number of network nodes, the types of traffic (based on the data associated with the various artifact types being transmitted through the network in support of the movement through the battlefield), and the volume of traffic—at any given time during the lifetime of the mission.

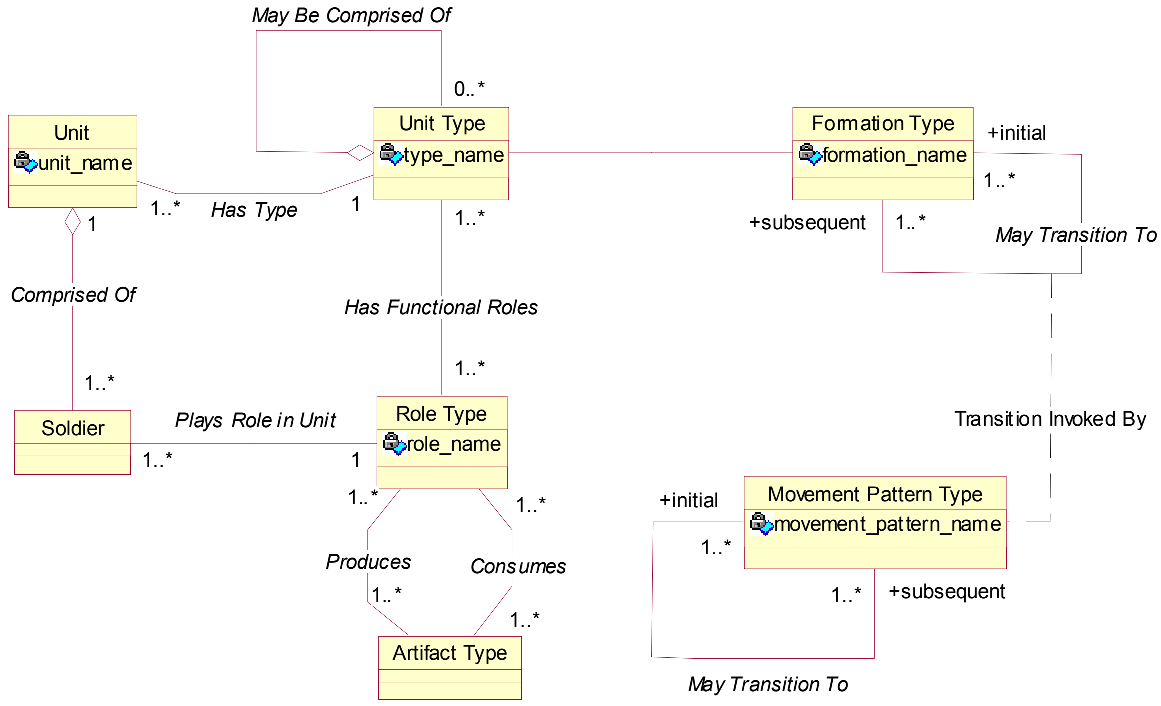


Figure 25. The core relationships of the domain.

As shown in the figure above, we can relate the role-playing of members of the fighting force (and their implied responsibilities) to the types of artifacts and, ultimately, to the relationship each soldier "role player" has to the artifacts in the inventory. In addition, we are able to establish relationships between the type of fighting force unit with the types of formations that are defined for that unit type. Since formations are static, but precede and follow one another as a result of transitions made during the movement on the battlefield, we can define a "ternary" relationship between two allowable adjacent formation types and the movement pattern type in which the two formations are part.

In other words, the fighting force adopts patterns of movement on the battlefield as a result of mission planning and as a response to conditions in the field. The Army, through its experience, has identified formations that maximize flexibility, lethality, and security of the fighting force, and has defined a set of movement patterns that can easily be used to transition the fighting force between various formation types while moving on the battlefield. Just as with "hive-oriented" insects such as ants and bees, the fighting force can be thought of as an "organism" whose parts move in carefully choreographed "dances". However, unlike insects, military planners have carefully worked the effectiveness of these "dances" for years with intent of maximizing objectives in the field of battle. It is not an accumulation of random movements transmitted through inheritance and natural selection.

Therefore, given this "strongly typed" model structure, we have subjected the properties of the model to scrutiny for the purpose of creating a quasi-formal mapping of these characteristics to an appropriately defined network model capturing those properties essential to exploring the impact of environmental effects on the QOS of the wireless LAN.

4.4 Modeling the Roles and Use Cases of the Fighting Force

The Use Case model depicted in Figure 21 provides an inventory of the core "actors" and domain "transactions" that are deemed relevant in the modeling of the fighting force on the battlefield. This abstract model consists of three role-based "actors": Command, the Fighting Force, and the Enemy Force. The analysis model to be created involves transactional exchanges involving these three abstract entities. These entities engage one another through the indicated Use Cases: Pass Situational Awareness Data, Receive Fragmentary Orders (FRAGOs), Engage Enemy, Seize Objective, Enemy

Counterattack, and Defend Objective. These abstract transactions constitute the scope of inventory of concepts that have to be modeled in order to devise an acceptable model for analyzing and exploring the wireless network. We next drill into each of these Use Cases through the use of Sequence Diagrams, and briefly discuss the context of these transactions. We will tie the details of this analysis model to the set of assumptions defined for input to constructing a network simulation model in the next chapter.

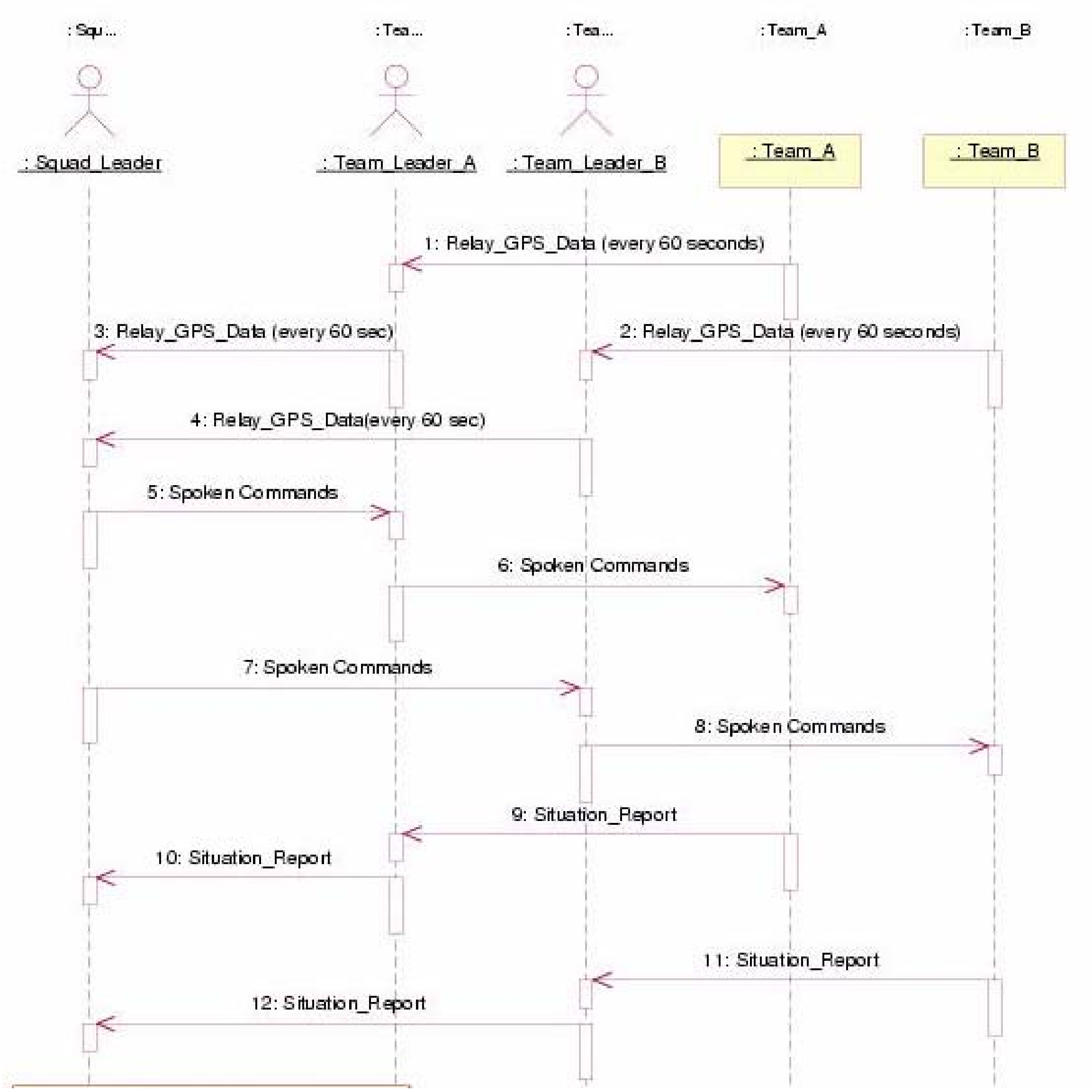


Figure 26. The Pass situation awareness scenario.

The spoken commands/situation report sequence will iterate until the unit transitions to the receive FRAGO or Engage Enemy use cases. The situation report is based on a soldier seeing something that may indicate a threat to the force—such as sighting an enemy, booby traps, landmines, etc.—and reporting this information to an immediate superior in the unit. It can also be a report of passing a significant landmark or graphical control measure (in regards to the map overlay) to mark the unit's progress towards the mission objectives.

While shown as a separate Use Case, the Pass Situational Awareness Data scenario is a continuous process. We have separated it out to show how the process occurs during normal unit operations. Location data (GPS) is relayed throughout the unit and higher every 60 seconds. The leaders issue spoken commands to their subordinates using the VOIP function and subordinates can pass up Situation Reports either verbally or via email. Soldiers generate a Situation Report when they see something that may be a threat to the force. The soldier can also report on passing a significant landmark to mark the unit's progress. The Spoken Commands/Situation Report sequence will iterate until the unit transitions to the Receive FRAGO or Engage Enemy use cases.

When a unit receives a FRAGO, it is passed down through the leaders to the soldiers. The FRAGO may have text or additional map graphics that mark a change to the base OPORD. The leaders at each level will extract the information their own unit requires and distribute that information to their subordinates. The subordinate review the order and, if they have questions or problems, send back a Request for Information (RFI).

Unit leaders at each level can either answer the RFI themselves or pass it up to the next level. RFI Resolutions return down through the chain of command. The RFI/RFI Resolution process can iterate until the subordinate feel they have enough information to execute the mission or until senior leaders prevent more questions. At this point, the soldiers will notify the leaders that they understand the order.

At some point during the mission, the squad will make contact with the enemy. The soldiers in the fire team will Report Contact, usually via a SALUTE report. The

SALUTE report is a standard means of reporting on the Size, Activity, Location, Unit designation, Time, and Equipment of any enemy force encountered on the battlefield. Once the squad leader receives the contact report, he will issue orders on how the fire teams should react to the enemy. The fire teams respond with situation reports to keep the squad leader informed of the situation. The Issue Order/Situation Report process continues until there is a resolution to the conflict.

If the squad is successful, they will seize the objective. The squad leader issues a force arrangement order with instructions on where the soldiers should place themselves to guard against an enemy counterattack. The teams pass up ACE reports, which are standard reports about the Ammunition, Casualty, and Equipment status within the teams. Using this information, the SL will issue an order on how to distribute the remaining resources among the members of the squad.

When the enemy counterattacks and the squad defends the objective, the sequence is the same as with the Engage Enemy sequence. Again, the Issue Order/Situation Report sequence will iterate until there is a resolution to the conflict.

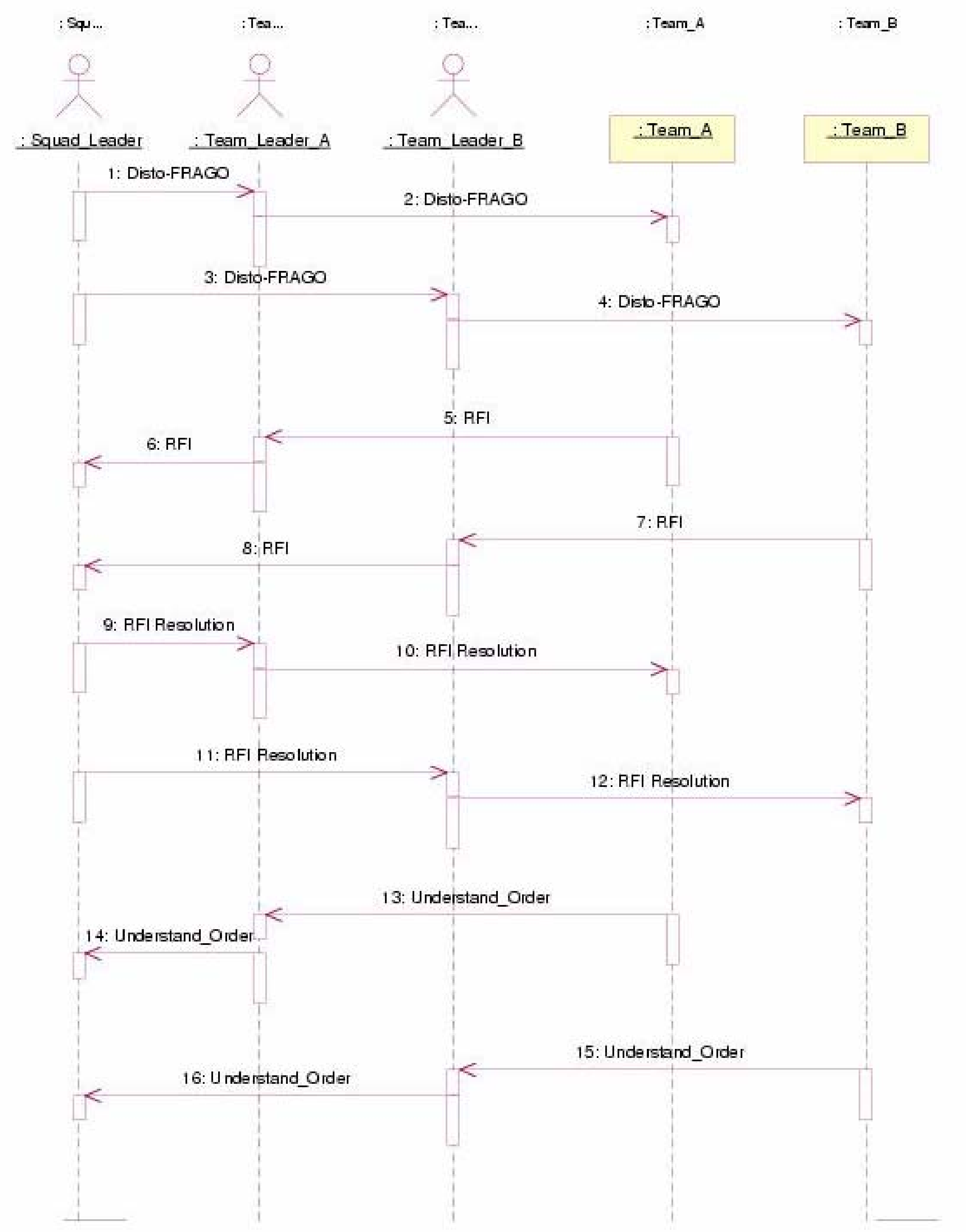


Figure 27. The Receive fragmentary orders scenario.

A fragmentary order (FRAGO) can have text explaining changes to a previous order, as well as new graphical content derived from map overlays. The RFI/RFI Resolution sequence can iterate until the subordinates feel they can perform the mission, or until senior personnel prevent more questions. The squad leader can pass the RFI up through the chain of command or answer the question himself.

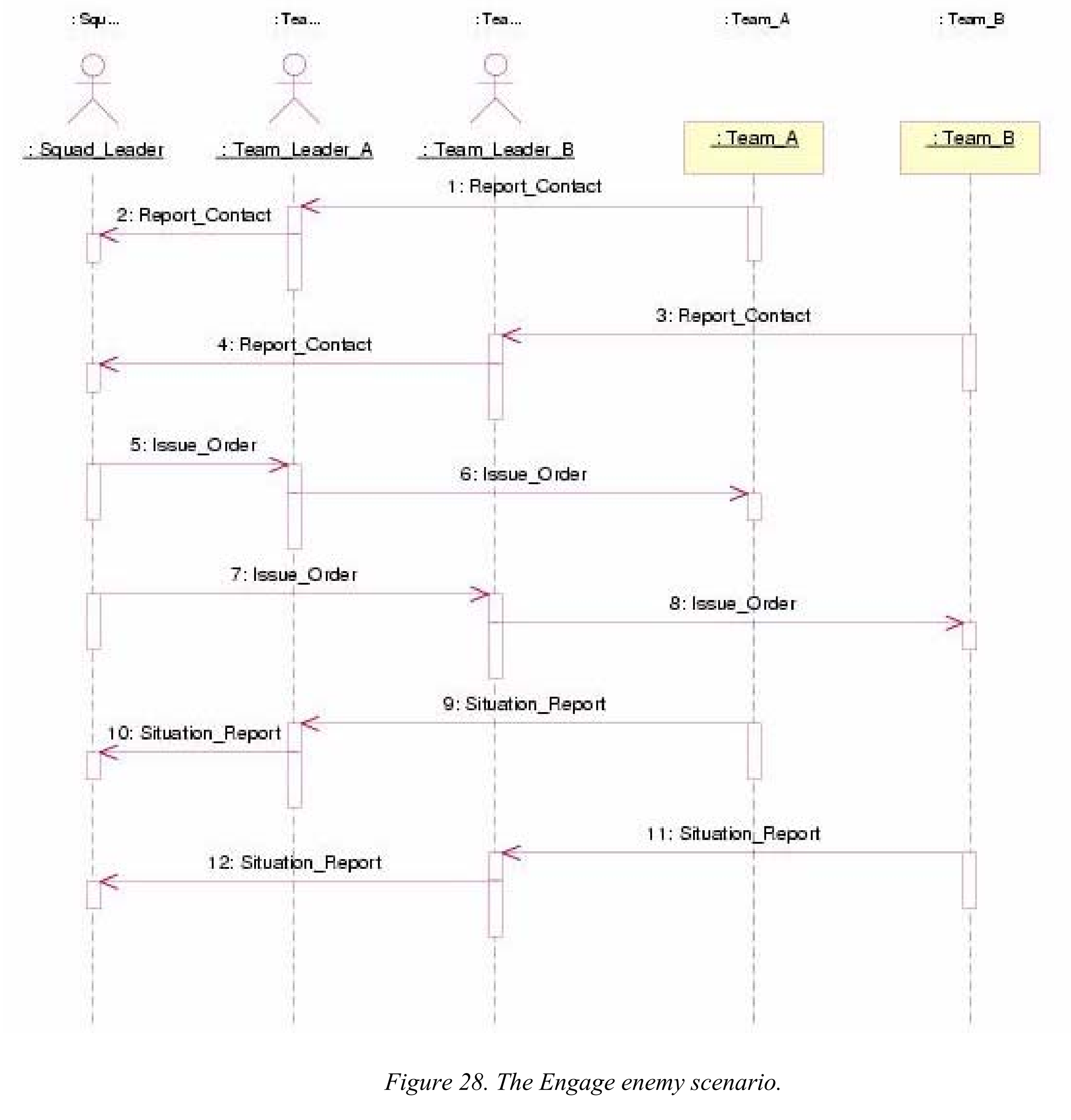


Figure 28. The Engage enemy scenario.

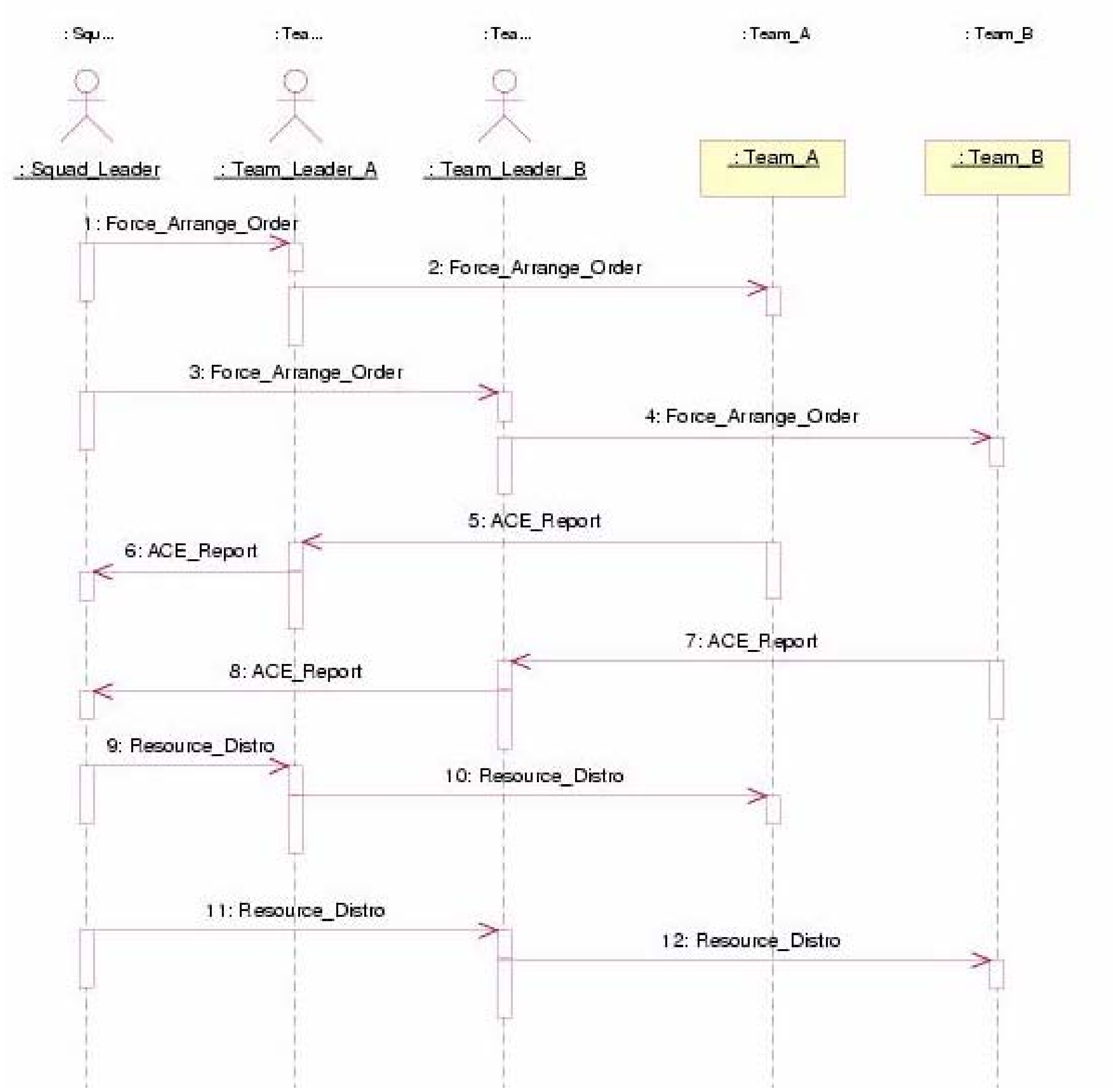


Figure 29. The Seize mission objectives scenario.

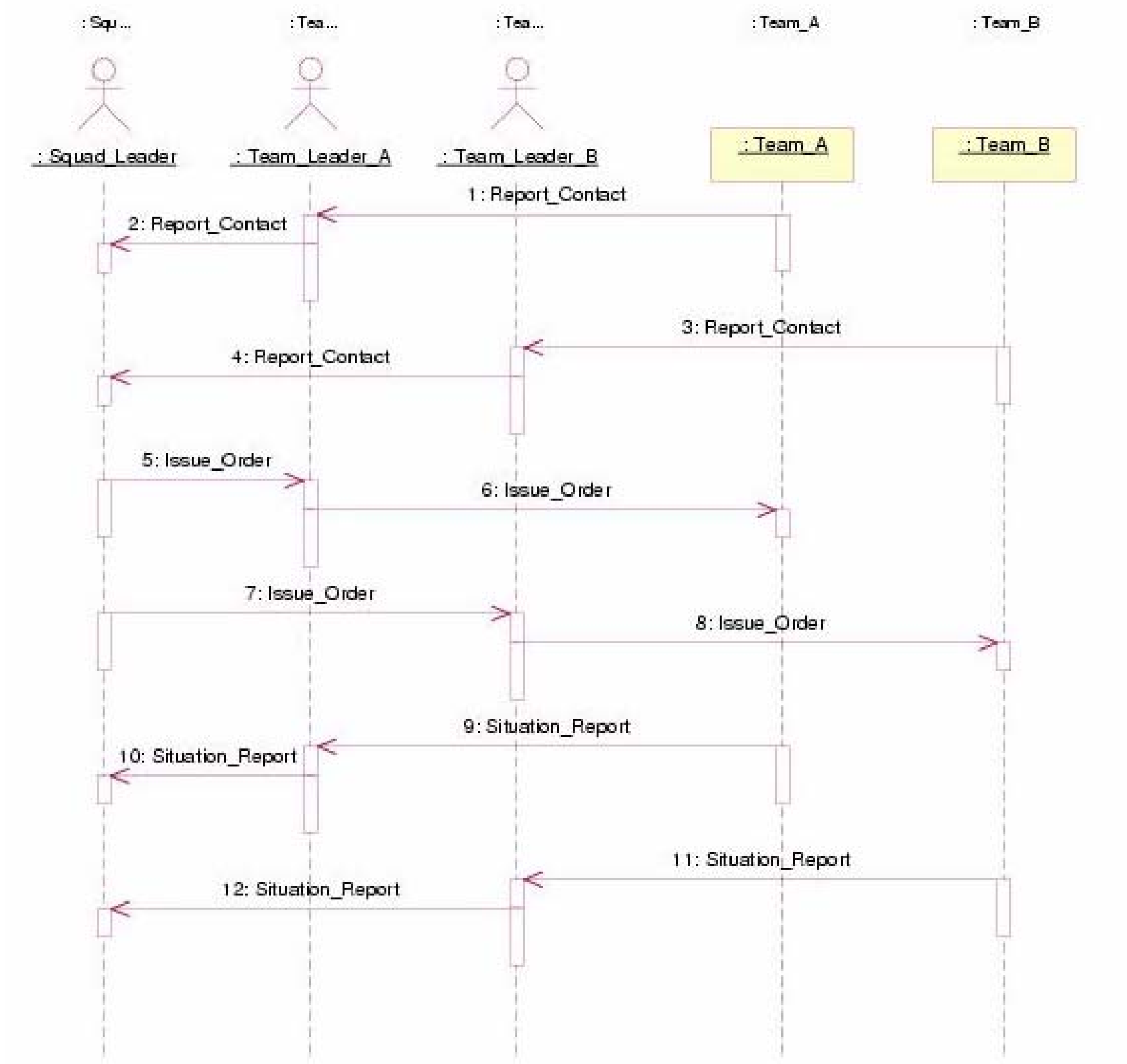


Figure 30. The Enemy counterattack scenario.

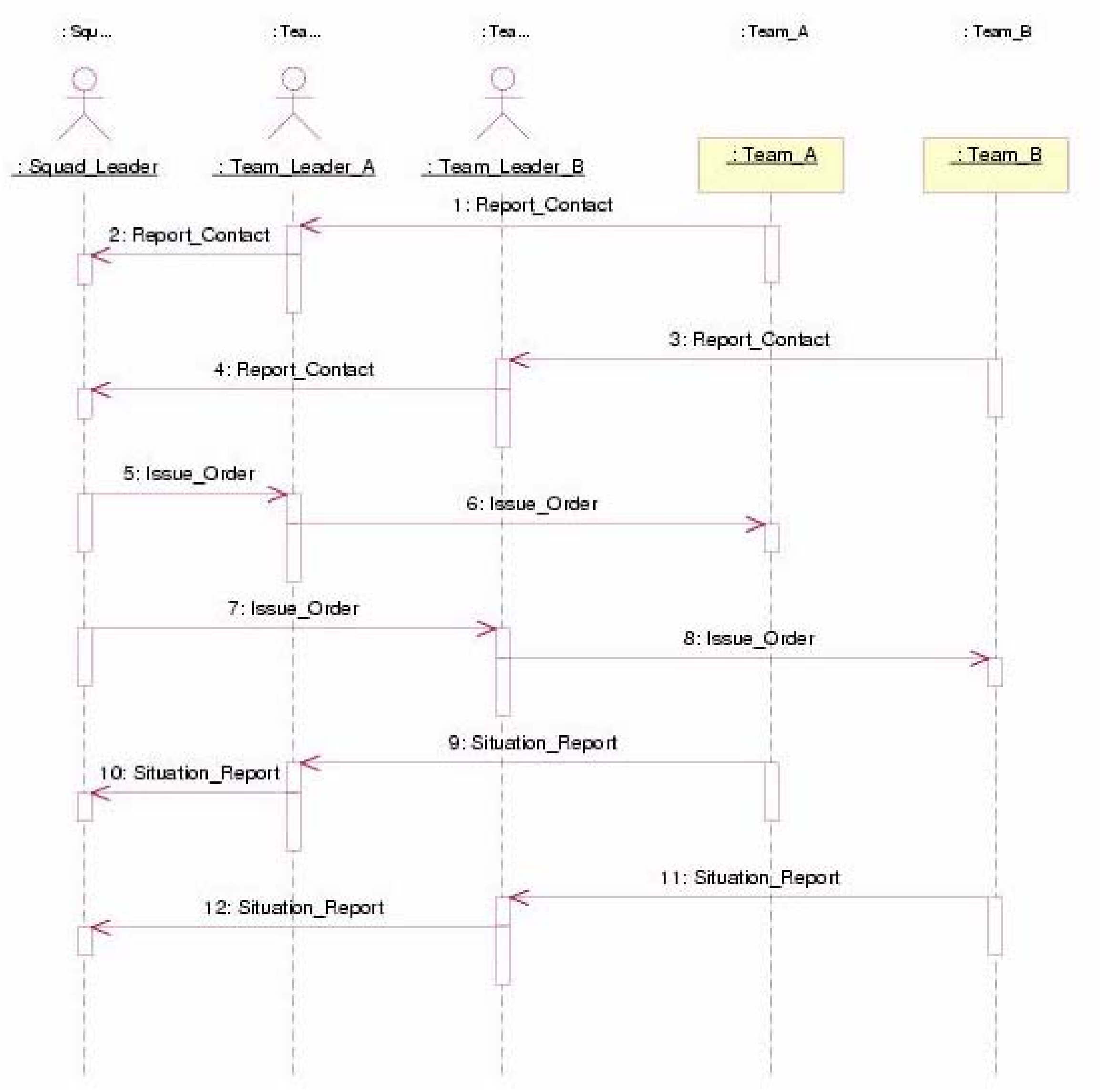


Figure 31. The Defend objectives scenario.

To create an accurate network model for simulation, we had to understand how unit movement, fighting force roles, and artifact creation and distribution affect network traffic in terms of traffic pattern and traffic timing as the mission unfolds through its stages. As will be shown in Chapter 5, the simulation engine allows us to define various node and network traffic parameters to accurately model our network. Understanding unit movement, fighting force roles, and artifact creation and distribution contributes to our understanding of how to set these parameters to accurately define the model in software. This understanding also allows us to determine expected network traffic as a result of document artifacts and the effect on network QoS as a result of the patterns of movement, which helps us to determine the validity of our results.

4.5 Modeling the Formations and Patterns of Movement

We have the basic structure of the domain model, but need further drill-down into the domain semantics to be able to define a mapping transition to the network simulation model. This drill-down involves understanding the detailed compositional structure of a mission scenario—as abstracted in the previous Sequence diagrams—into its constituent conceptual elements. It is these elements that will support the instantiation of a particular fighting force in the battlefield.

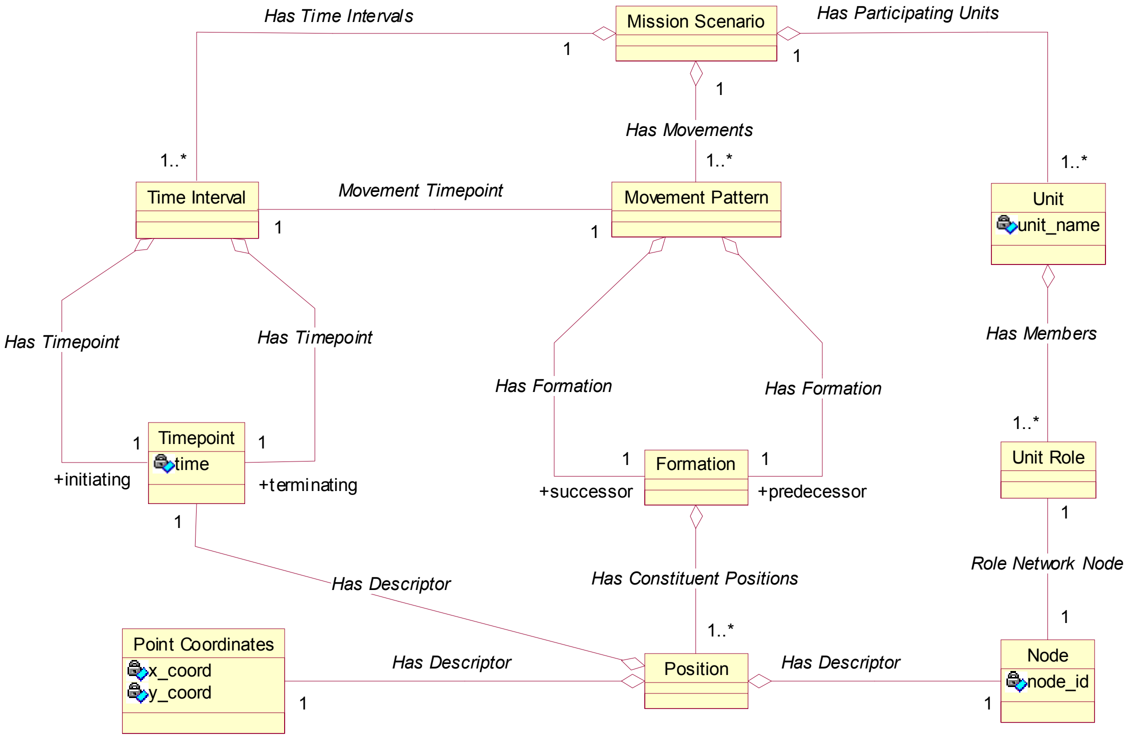


Figure 32. The mission scenario decomposition hierarchy.

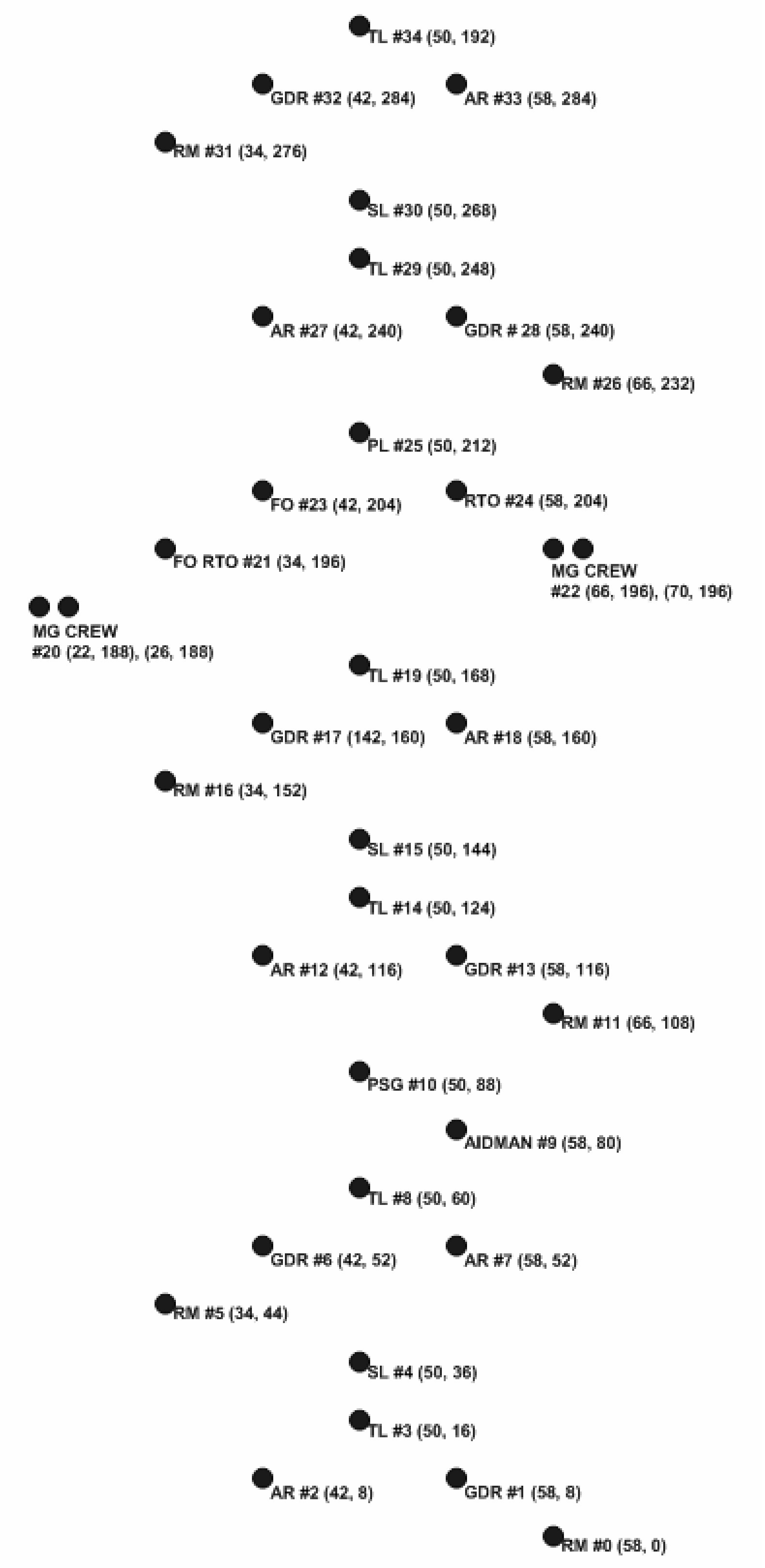


Figure 33. An instantiated formation, with nodes, positions, and roles.

From the UML Class diagram, we see the decompositional nature of a scenario (which directly corresponds to a sequence of interactions associated with a Use Case), comprised of the patterns of movement, each with a starting and ending formation. Each formation consists of a number of positions, consisting of a tuple of (Node, Coordinates, Timepoint). A node is associated uniquely with a role that is part of a specific unit in the fighting force. This leads us in a position to derive the entire sequence of the scenario from the assembling of the constituent positions adopted by each node in the units involved in the scenario, tracking each unit as it move through the battlefield space, progressing through the formations and the patterns of movement.

From this semantic model, we are able to state the complex orchestration of movement as a sequence of individual movements by the nodes as they progress through the scenarios. This is exactly the representation required to capture the node movement for purposes of simulation in *ns-2* (as discussed in the next chapter). As shown in the node diagram on the previous page, we have the roles, node identifiers and positions occupied by the nodes associated with the unit's roles. In this example, we show the platoon formation, with its 3 squads (forward, center, and trail) aligned in the platoon column formation. This node diagram represents a snapshot of the progression of the scenario at one time point. Onto this base representation, we need to layer the succession of time points comprising the scenario, as well as some designation of the specific network traffic at the time points.

4.6 Meta-level Mapping Between the Domain and Simulation Models

A key aspect of the analysis component of this research has concerned how to map between a domain model representing the fighting force to a model appropriate for use in constructing network simulation scripts in *ns-2*. The primary issues here are as follows: (1) moving from a perspective of modeling the enterprise-oriented aspects of the fighting force to one based on considering the fighting force as a collection of network nodes with certain traffic and movement properties, and (2) translating the enterprise semantics of entities, relationships and static constraints to those appropriate for creating topological models useful for simulating network dynamics.

To make this transition, we considered a number of ways to represent the model but, ultimately, the process consisted of the following: (1) understanding the semantics of the *ns-2* simulation model (described in detail in the next chapter); (2) creating a meta-model of these semantic concepts and relations as a UML Class diagram; and (3) using this understanding of the semantics of *ns-2* to dictate creating a set of mapping policies between the domain and the simulation models. Since one of the considerations of this research problem has been to create a simulation model that could be extended by creating specific battlefield scenarios, it was important to be able to make the transition between the models as straightforward as possible. It is our belief that this process could be automated with an appropriately-constructed model capture and compilation tool, possibly as an extension to Rational Rose® through the use of COM component add-in technology (which is discussed further in the final chapter of this research thesis) [RAT98].

CHAPTER 5

PRESENTATION OF THE SIMULATION MODEL OF THE WLAN

5.1 Network Simulation in the *ns-2* Environment

The *ns-2* simulator is an object-oriented, discrete-event simulator developed to test various networking protocols over wired and wireless networks. *Ns-2* is written in C++ with an Object Oriented Tool Control Language (Otcl) interpreter as the simulator front end. *Ns-2* uses both languages to satisfy two necessary but opposed qualities of performing network simulations. On the one hand, the simulation requires efficient manipulation of bytes and packet headers and implementation of algorithms that run over large data sets.

Therefore, *ns-2* uses C++ to implement detailed protocols requiring fast run times, but infrequent changes. On the other hand, the researcher will want to make many little changes to the simulation to determine their effect or explore a number of different scenarios. Scripting the simulation in Otcl allows quick, interactive changes to the simulation script, but runs much slower than C++. *Ns-2* links these two languages to allow quickly written simulation scripts and fast run times of the network protocols [KFV00].

By the very nature of it's split level programming, *ns-2* offers several useful advantages for this project. First, we were able to quickly write simulation scripts that scaled up the network from fire team to platoon level. Second, we could customize our traffic sources to model the different traffic types generated by the LW system. Finally, we could introduce network dynamics into the simulation to model lost nodes or links. More importantly, when we decided to make changes to the MAC protocol, we could easily create a new IEEE 802.11b MAC protocol within the C++ source code. After creating and compiling the new protocol, we could include the new protocol in our simulation scripts and quickly analyze the results.

5.2 Discussion of the Simulation Model

In describing the software simulation model, there are three relevant aspects; the base simulation script, the node movement script, and the network traffic script. The base simulation script manages the nodes, manages the trace files, and extracts and displays packet information from the traffic sinks. The node movement file establishes an initial node position, provides direction and speed for node movement and defines network path information. Finally, the network traffic script establishes agents and applications at each node to simulate traffic on the network. We combined these scripts to simulate the LW WLAN network of a light infantry platoon on an attack mission.

For this scenario, we chose to simulate a LW equipped, light infantry platoon conducting an attack mission. This mission is very simplistic; however, it contains enough detail to adequately capture how the network changes as the platoon maneuvers across the battlefield. For example, it contains each of the three movement techniques, of which the traveling overwatch and bounding overwatch techniques require to the platoon members (and thus the nodes) to change the distances between themselves. In addition, The scenario consists of five steps, each simulated separately by using different node movement files and slightly different base scenarios. The platoon starts in a platoon column formation, as shown in Chapter 2. We started with the rear soldier (the RM in the trail team of the trail squad) as the base from which we built the formation. This soldier started at position (100, 0) along an x-y axis. By locating each node 8 meters along the x-axis and 8 meters along the y-axis, we ensured approximately 10 meters between soldiers. In this step, the platoon uses the traveling movement technique, which requires 20 meters between the squads. To complete this step, the platoon moves forward 200 meters at 0.5 m/s. We chose that speed based upon the writer's experience in various Army field exercises. We also chose a distance of 200 meters because it gives the network enough time to stabilize and a longer simulation time would not add value to this particular simulation.

Step 2 places the nodes in an initial position equivalent to their final position from Step 1. The platoon transitions to traveling overwatch by having the lead squad move forward 30 meters. The squad also transitions to traveling overwatch by increasing the distance between its teams to 50 meters. The remaining squads and the HQ team continue to use the traveling movement technique. Figure 4-2 shows the node positions after this transition. Finally, the platoon moves forward another 200 meters at 0.5 m/s.

As in Step 2, initial node position in Step 3 is determined by the last node position in Step 2. At the beginning of Step 3, an external event triggers the platoon to move into an assault position. The PL directs the 1st (lead) squad to form a squad line to the left of the original direction of travel. One of the MG crews joins the 1st squad and takes a position online and the left of the 1st squad. After the 1st squad is in position and overwatching the enemy, the 2nd (center) squad forms a squad line to the right of 1st squad. The second MG crew positions itself online and to the right of 2nd squad. The PL, RTO, FO, and FO RTO take positions to the rear of 1st squad. The PSG and Aidman place themselves just behind the 2nd squad. After the 2nd squad is in position, the 3rd (trail) squad is directed to an assault position 75m to the right and 50m ahead of 1st and 2nd squads. During this maneuver, we increased the soldiers' speed to 1 m/sec. Figure 4-3 shows the final positions of the platoon after this step.

In Step 4, the 3rd squad assaults across the objective at 1 m/s by moving from right to left (with respect to the 1st and 2nd squads) to a point about 25 meters to the left and 50 meters ahead of the 1st and 2nd squads. Figure 4-4 shows the final node positions after Step 4. Step 5 finishes the scenario by having 1st and 2nd squads move forward and form a defensive perimeter with 3rd squad. The HQ team assumes a position in the center of the perimeter. Figure 4-5 shows the final node positions at the end of the scenario.

The first step in writing the Tcl script to match our scenario is to write the node movement files. These files contain only the information required to establish an initial node position, provide movement instructions, and next hop information. Mobile nodes are assumed to move in a three dimensional (X-Y-Z) topology, however, the *mobilenode* is assumed to move in a flat plane since the third dimension is not used. We wrote each step into a separate movement file, in which we explicitly defined the node starting position, the node's destination and the node's speed. The code in Figure 4-6 is an example of how to define node movement. After we defined the node movement, we had to append the next hop information required by the General Operations Director (GOD) object. The GOD object stores the number of *mobilenodes* and a table of shortest number of hops required to reach from one node to another. It is possible to compute this information during the simulation, however that can be time consuming [GRM00]. So, we used the `calcdest` tool provided with the *ns-2* distribution to compute next hop information and append the appropriate code to our node movement files. The appended code provides initial next hop information and any changes to the GOD object at the beginning of the simulation. Figure 4-7 provides some examples.

```
Initial node placement
$node_ set X_ <x1>
$node_ set Y_ <y1>
$node_ set Z_ <z1>

Defining node movement
$ns_ at 10 "$node_ setdest <x2> <y2> <speed>"
```

Figure 34.

```
Setting Next Hop Information
$god_ set-dist <src> <dest> <#hops>

Changing Next Hop Information
$ns_ at <time> "$god_ set-dist <src> <dest> <#hops>"
```

Figure 35.

After defining the node movement files, we defined the traffic pattern file, which remains the same for all steps in the scenario. In LW, each platoon is on the same wireless channel, therefore, every soldier in the platoon receives packets from every other soldier [HSM02]. To define the traffic pattern, we created UDP and Loss Monitor agent at each node and attached a CBR traffic generator to each of the UDP agents. However, *ns-2* does not provide a means within the Otcl code to set the agent to broadcast the CBR traffic to all of the other nodes. Admittedly, wireless packets are, by the very nature of wireless communications, broadcast to every other node within range of the transmitting station. However, within *ns-2*, the agents are only set up to provide unicast links [KFV00]. Therefore, we had to develop a way to model broadcast communications.

Our setup for a broadcast communications is best described as a two dimensional matrix, as shown in table 4 below. The source agents are shown along the left side and the sink agents are shown along the top. Each square with a dot represents a connection between one source on one port at one node to one sink on one port on another node. As shown in the diagram, there are no connections where the source index and sink index are equal, therefore, since there can be no connection between a source and a sink at the same node. The *ns-2* simulator handles this by having each node maintain a list of 34 source agents and 34 sink agents. Each agent is assigned a separate port on the port demultiplexer, through which it communicates with its assigned distant end. Since we created a two-dimensional matrix of sources and sinks, it is a simple matter to iterate through the matrix and create the links between the agents and assign the traffic generator to the source agent.

	sink(0)	sink(1)	sink(2)	sink(3)	sink(4)	sink(5)	sink(6)	sink(7)	sink(8)	sink(9)
src(0)		●	●	●	●	●	●	●	●	●
src(1)	●		●	●	●	●	●	●	●	●
src(2)	●	●		●	●	●	●	●	●	●
src(3)	●	●	●		●	●	●	●	●	●
src(4)	●	●	●	●		●	●	●	●	●
src(5)	●	●	●	●	●		●	●	●	●
src(6)	●	●	●	●	●	●		●	●	●
src(7)	●	●	●	●	●	●	●		●	●
src(8)	●	●	●	●	●	●	●	●		●
src(9)	●	●	●	●	●	●	●	●	●	

Table 4. Source-sink connection matrix for the ns-2 model.

```
#create UDP sources and LossMonitor sinks at each node
#connect each node to every other node
for {set i 0} {$i<$val(nn)} {incr i} {
  for {set j 0} {$j<$val(nn)} {incr j} {
     if {$i != $j} {   ;#Doesn't allow connection b/n source and sink
at same node
        set agList [$ns_ create-connection-list UDP $node_($i)
LossMonitor $node_($j) 0]
        set src($i.$j) [lindex $agList 0]
        set fid_ $i.$j
        set sink($i.$j) [lindex $agList 1]
     }
  }
}
# Create CBR traffic generators for each node
# Connect the traffic generators to each source port on each node
for {set i 0} {$i<$val(nn)} {incr i} {
  for {set j 0} {$j<$val(nn)} {incr j} {
     if {$i != $j} {
        # Create GPS Application
        set gps($i.$j) [new Application/Traffic/CBR]
        $gps($i.$j) set packetSize_ 185
        $gps($i.$j) set interval_ 60000ms
        $gps($i.$j) attach-agent $src($i.$j)
        $ns_ at [expr $i.$j/1.0] "$gps($i.$j) start"
        # Create email Application
        set email($i.$j) [new Application/Traffic/CBR]
        $email($i.$j) set packetSize_ 79
        $email($i.$j) set interval_ 400000ms
        $email($i.$j) attach-agent $src($i.$j)
        $ns_ at [expr 60 + ($i.$j/1.0)] "$email($i.$j) start"
        # Create overlay application
        set overlay($i.$j) [new Application/Traffic/CBR]
        $overlay($i.$j) set packetSize_ 231
        $overlay($i.$j) set interval_ 10000ms
        $overlay($i.$j) attach-agent $src($i.$j)
        $ns_ at [expr 120 + ($i.$j/1.0)] "$overlay($i.$j) start"
     }
  }
}
```

Figure 36. Traffic connection tcl script.

We wanted to create traffic generators that matched actual traffic patterns as closely as possible. So, we used data generated by a LW field test at Fort Polk, Louisiana and subsequently subjected to statistical analysis by researchers at the United States Military Academy at West Point [HSM02]. The research at West Point revealed four major types of packets used in LW: Voice over IP, Active Soldier (consisting of GPS information), Email, and map overlay packets, all discussed in Chapter 4. We created CBR traffic generators for each type of packet and used the derived inter-arrival times and packet sizes from the West Point research. Table 5 shows the average inter-arrival times and packets sizes we used for our research

Application Type	Packet Size (byte)	Interarrival Time (sec)
Email	78	424
GPS	185	60
Overlay	231	10
VoIP	106	0.170

Table 5. Application Configuration Parameters

However, using all four traffic generators generated an extremely large number of events for the script to record in the trace files. As an example, we can show the number of events generated using our connection setup with just the GPS traffic generator sending a packet every 60 seconds. For all 35 nodes, there are 34 traffic generators for a total of 1190 traffic generators. Additionally, there are 1190 traffic sinks. Every 60 seconds, all the traffic generators will generate a packet causing a send event to be recorded. A receive event will also be recorded when the packet arrives at the sink. Therefore, there are 2380 events generated by the simulated action of each node sending one packet. Over the life span of a 400 second simulation, this translates into 14,280 events (400 sec/60 sec ≈ 6 send events/node where 6 events * 2380 simulation events = 14,280 simulation events). This number is just for events at the agent level and does not include events at the router or MAC layers, which can include Address Resolution Protocol packets and DSR protocol packets. The number also does not include any packet drop events. Therefore, we started this research using one traffic generator simulating GPS packets. However, we were able to determine that the VoIP application produced a majority of the network traffics, since this application broadcasts a 106 byte packets every 170 milliseconds. Because of this determination, we included Email and Overlay applications in our simulation. In addition, initial results indicated a large amount of packet drops deriving from a large number of DSR protocol and ARP packets broadcast by each node to determine routing and resolve MAC addresses. We wanted to allow the network to achieve a steady state before node movement, so we started each application at one-minute intervals, and started node movement one minute after the last application started. This process is very realistic as military missions will not start without a "communications exercise" to verify the working condition of each communications platform in the unit. For our scenario, we started the GPS application at time 0, which simulates the system being turned on and trying to find the other nodes in the network. The Email application starts at time 60 seconds, which simulates a function of check on the email application in the LW system. Finally, the Overlay application starts at time 120 seconds, thus simulation another function check. Finally, the nodes start movement, simulating a successful "communications exercise" and the start of the mission. Figure 36 shows how we created these applications.

```
Define the node options
set val(chan)          Channel/WirelessChannel  ;# Channel
set val(prop)          Propagation/TwoRayGround ;# Radio wave model
set val(netif)         Phy/WirelessPhy          ;# Physical network
interface
set val(mac)           Mac/802_11               ;# MAC Protocol
set val(ifq)           Queue/DropTail/PriQueue  ;# Type of Queue
set val(ll)            LL                       ;# Link Layer type
set val(ant)           Antenna/OmniAntenna      ;# Propagation
direction
set val(ifqlen)        50                       ;# max packet in ifq
set val(adhocRouting)  DSR                      ;# Ad-hoc Routing
protocol

Set the Parameters
$ns_ node-config -adhocRouting $val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propInstance [new  $val(prop)] \
                 -phyType $val(netif) \
                 -channel [new $val(chan)] \
                 -topoInstance $topo \
                 -agentTrace OFF \
                 -routerTrace OFF \
                 -macTrace ON
```

Figure 37.

The final step in writing an OTcl script to match our scenario is to create the base script. In this script, we create the *mobilenodes* objects, set their parameters and set up traces. We used the code in Figure 4-10 to create the nodes and set their parameters.

We've set the channel, propagation, network interface, interface queue and link layer type options to the basic settings *ns-2* requires for wireless nodes [KFV00]. The MAC protocol option is obviously set to 802.11. While LW actually uses the On-Demand Multicast Routing Protocol (ODMRP), we are using Dynamic Source Routing (DSR) as the ad-hoc routing protocol. ODMRP is not a defined routing protocol within *ns-2* and DSR is a reasonable approximation for our purposes [JOD96][LSG00]. After we configured the mobile nodes, we used the code in Figure 37 to actually create the mobile node objects. When the mobile node object is created, a routing agent is created as specified and the network stack is created and connected to the channel.

```
for {set i 0} {$i < $val(nn) } {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0          ;# disable random motion
}
```

Figure 38.

In addition to defining the *mobilenode* characteristics, we opened the ns and nam (network animator) trace files and instantiated the ns and nam trace objects as shown in Figure 38. The ns trace object traces all packet events that occur in the simulation at the agent, router, and MAC levels. Packets events are defined as when a packet is sent, received, or dropped. We used the node configuration code (shown in Figure 39) to turn on or off tracing at the different levels as needed. We created awk scripts to filter these trace files and gather necessary statistics for plotting. Nam trace objects capture packet events as well as node movement events and write to a file that can be read by the nam utility to show an animation of the network.

```
# create trace objects for ns and nam
# Assingn file names to variables
set tracefd  [open step1-out.tr w]
set namtrace    [open step1-out.nam w]

# Use new trace format
$ns_ use-newtrace

# Create the ns trace object
$ns_ trace-all $tracefd

# Create the nam trace object
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

Figure 39.

CHAPTER 6

EXPERIMENTAL METHOD AND RESULTS

6.1 Selection of Performance and QoS Metrics

6.1.1 Packet drops organized by reason

For our experiment, we are interested in developing a metric the MAC layer can sense and act upon. One such metrics is the number of dropped packets. In *ns-2*, drops are classified by reason, therefore; we are interested in the packet drops that occur at the MAC Layer. Figure 1 gives a reason code used in the simulation trace file and the reason for the drop for those drop events of interest to us.

Reason Code	**Reason**
COL	MAC Collision
DUP	MAC Duplicate
ERR	MAC Packet Error
RET	MAC Retry Count Exceeded
STA	MAC Invalid State
BSY	MAC Busy
NRTE	RTR No route available
LOOP	RTR Routing loop
TTL	RTR Time to Live has reached zero
TOUT	RTR Packets has expired
CBK	RTR MAC Callback
DSR	Dropped by DSR Protocol
AREQ	Dropped by ARP

Table 6. Packet Drop Reason Codes

If packet drops are concentrated along one or two reasons, then we should get an indication as to what is happening in the network and an indication of network quality.

6.1.2 Throughput

We decided to use a quality of service metric to detect any link between the number of drops at the MAC and RTR level and network quality. In our simulations, throughput is a measure of how much real data, excluding routing and address resolution requests, is actually received by a node per second of simulation time. We looked at the total throughput for all stations sampled every ten seconds. This gives us an indication of how the quality of the network changes over the lifetime of the simulation. In addition, we should be able to see how changing the transmit radius of the nodes impacts network quality. For each packet an application sends, we calculated throughput by dividing the number of data bits sent and received by the elapsed time. The elapsed time is the time from when the sending application gives the the packet to the routing layer to the time when the receiving agent gets the packet from its routing layer.

6.2 Discussion of the Experimental Runs

For our research, we conducted three experiments where we changed the transmission radius for each node. The first experiment had a transmission radius of 350 meters; the second, 90 meters; the third, 30 meters. We achieve the different transmission radius by changing the Receive Power Threshold, which is a threshold value used to determine if a packet has enough power to be correctly received by a node. Table 7 shows the Receive Power Threshold for each transmission range we used. For wireless simulations in *ns-2*, transmitted packets have an associated power level. The simulator determines the distance between the sending and receiving nodes and uses that distance to calculate (based on the Propagation Model used) how much power the packet has remaining at the receiving node. If the remaining power is below the Receive Power Threshold, the packet considered as "not-received". Therefore, as we increase the Receive Power Threshold we effectively reduce the transmission radius of the nodes.

Transmission Range(Meters)	Receive Power Threshold(Watts)
350	9.5081×10^{-11}
90	2.1746×10^{-8}
30	1.7615×10^{-6}

Table 7 Receive Power Thresholds at each Transmission Range

By reducing the node transmission radius, our goal was to segregate the network into sub-networks based on unit organization. So, for a transmit radius of 350 meters, we have all soldiers in the platoon on competing for access to the network. For a transmit radius of 90 meters, we reduce the number of nodes competing for access to squad level and create three sub-networks. In turn, for a radius of 30 meters, we reduce the number of nodes competing for access to team level and create six sub-networks. Changing the transmission radius is, at best, an imperfect means of segregating the network into distinct sub-networks. However, we were able to reduce the number of neighbor nodes for any particular node, therefore, reducing the number of nodes competing for medium access. This should have the resulting effect of reducing the amount of dropped packets from MAC collisions, thereby increasing throughput.

6.3 Presentation of the Data

After our simulation runs, we discovered a majority of packets drops resulted from MAC collisions. Therefore, the first five plots (Figures 40-44) compare of the number of dropped packets caused by MAC collisions for the different transmission ranges for each step in the scenario. Each point in the graphs shows the total number of drops for all nodes in the previous 10 seconds. For all of these plots, we started plotting at time 120 since the first two minutes of simulation time are spent starting the agents. The simulation time from 0 seconds to 120 seconds shows a large amount of packet drops related to network setup, such as route discovery by the routing protocol and MAC address resolution required by the ARP table.

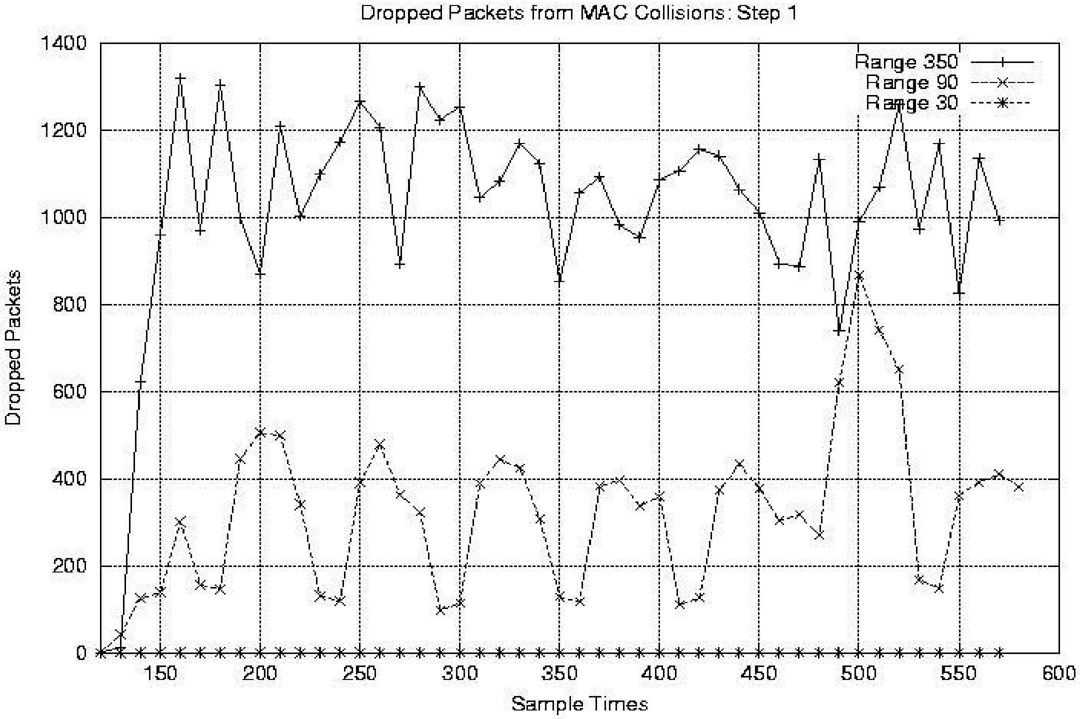


Figure40: Step 1 MAC collisions drop totals.

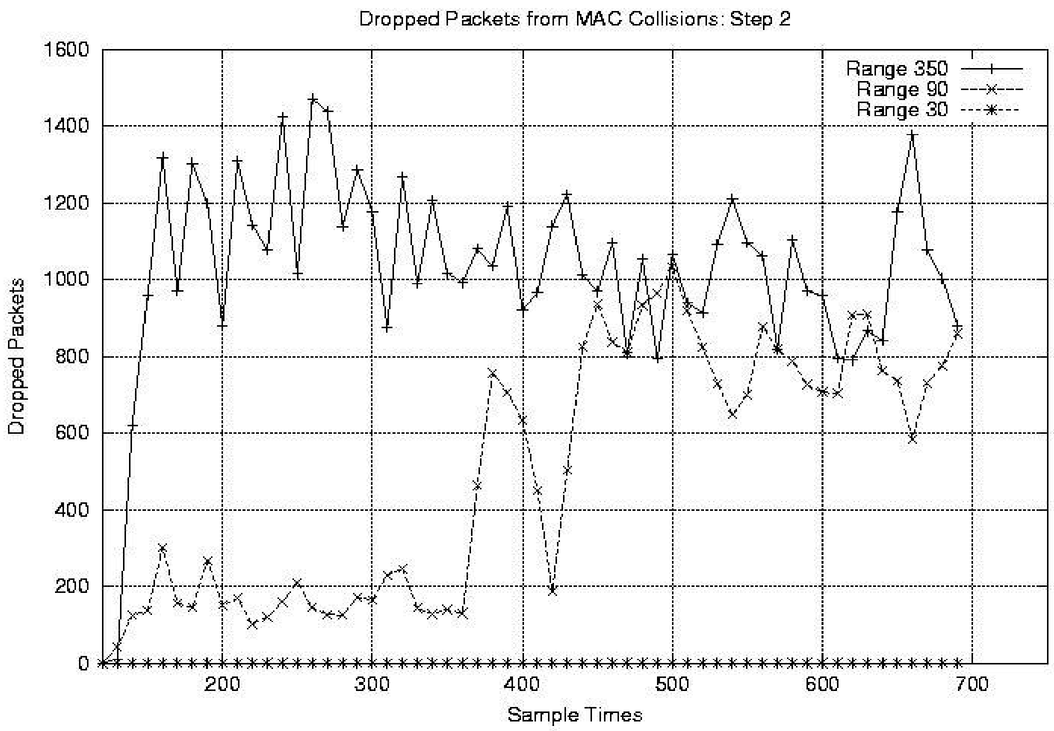


Figure41: Step2 MAC collisions drop totals.

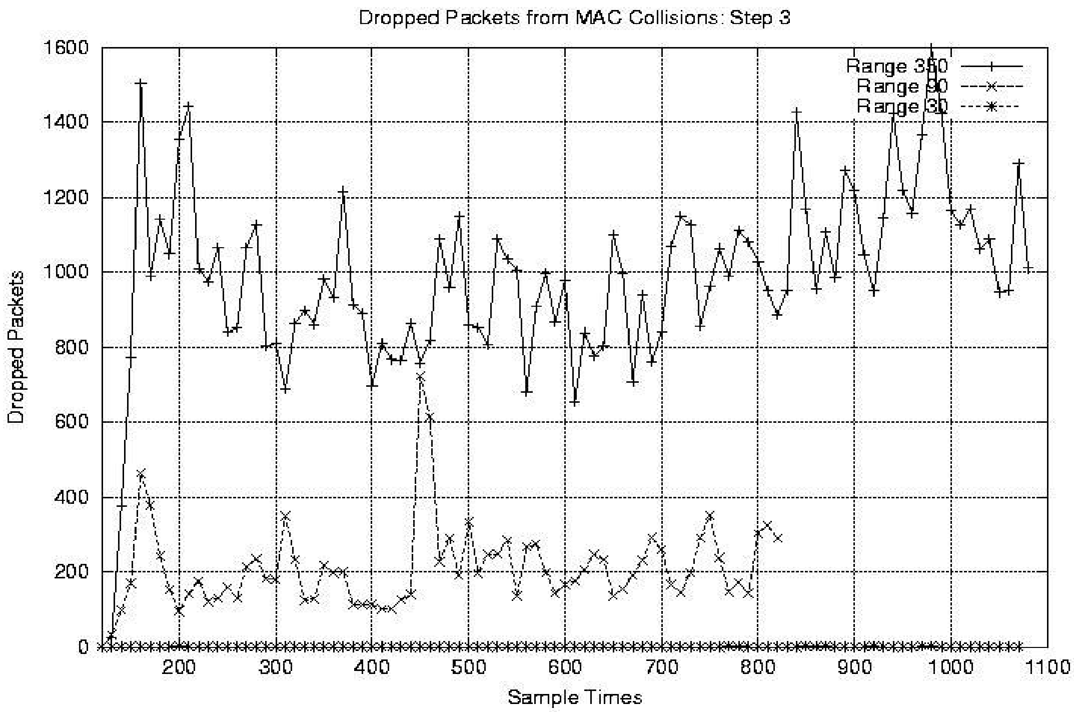


Figure 42: Step3 MAC collisions drop totals.

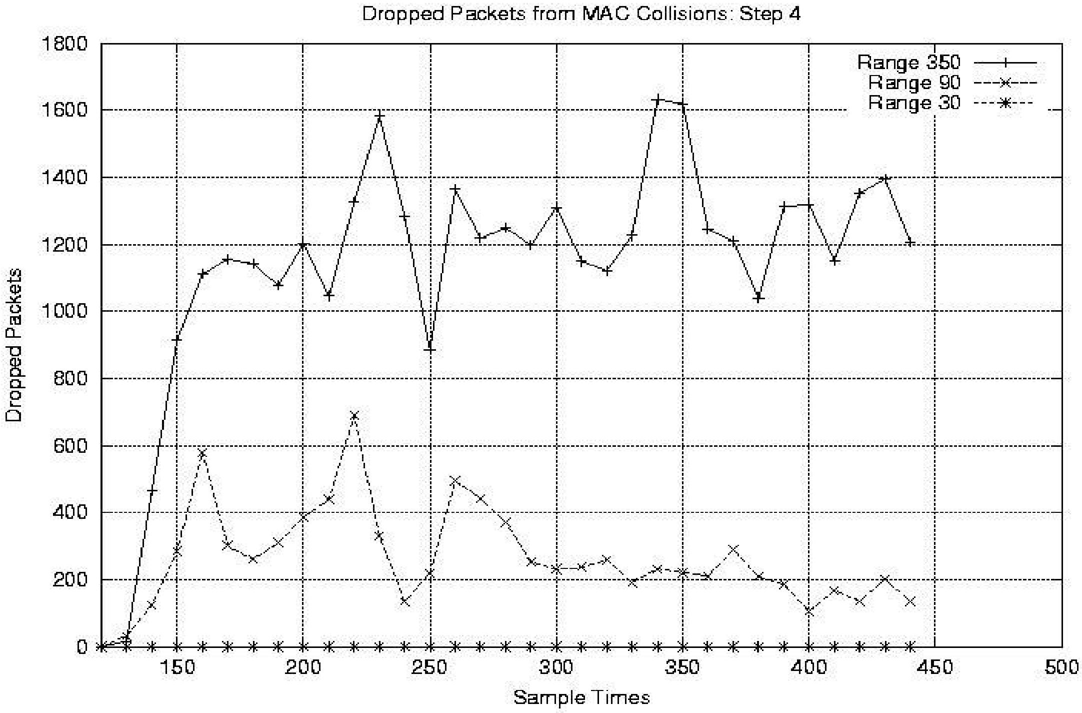


Figure 43: Step4 MAC collisions drop totals.

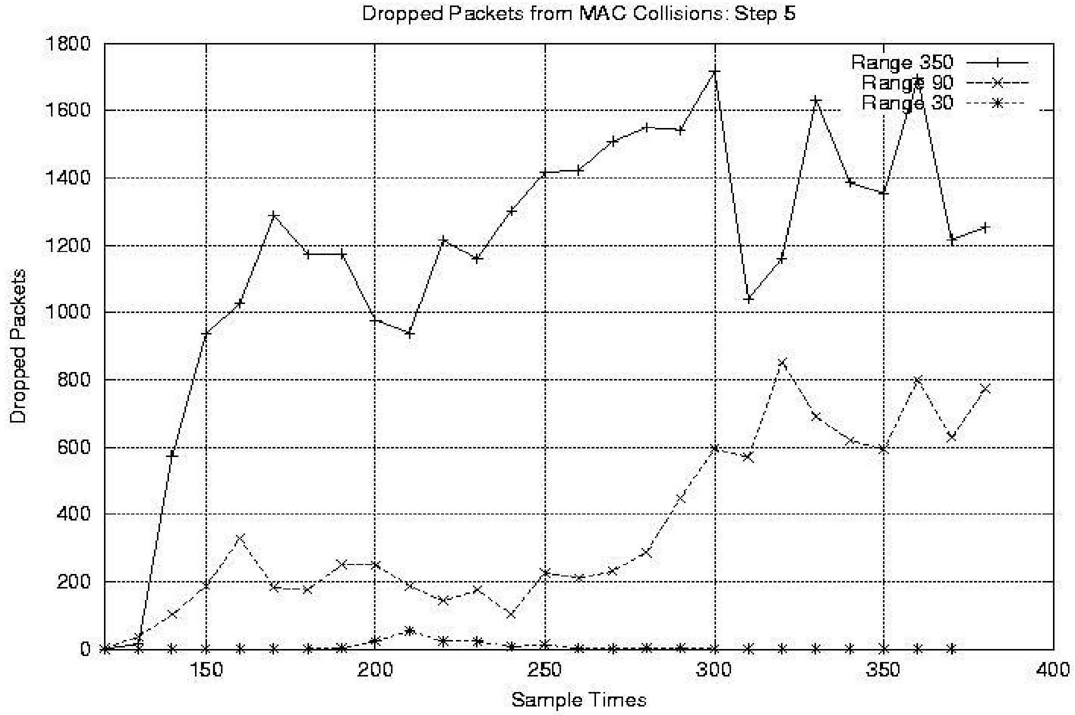


Figure 44: Step 5 MAC collisions drop totals.

Figure 45-49 show the number of drops from packets timing out for the scenario with a transmission range of 30 meters. This scenario did not have many drops due to MAC collisions. However, since the transmission range is so small, the network is physically segregated into distinct sub-networks. Therefore, packets with a destination outside the subnetwork will timout and be dropped by the node.

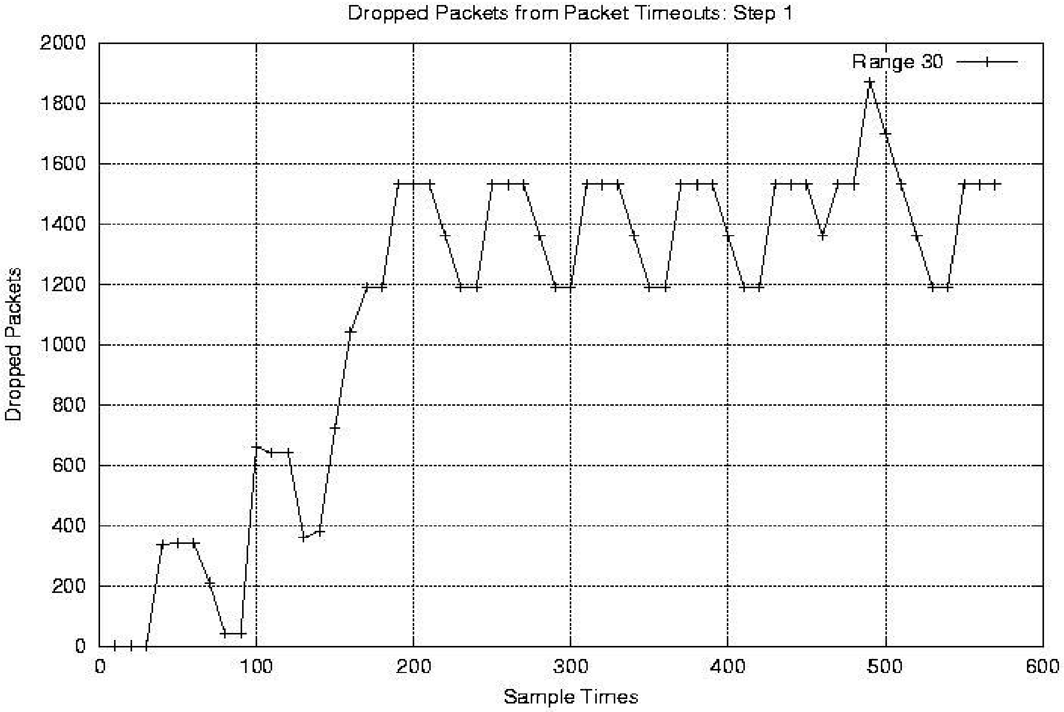


Figure 45: Step1 Packet timeout drop totals.

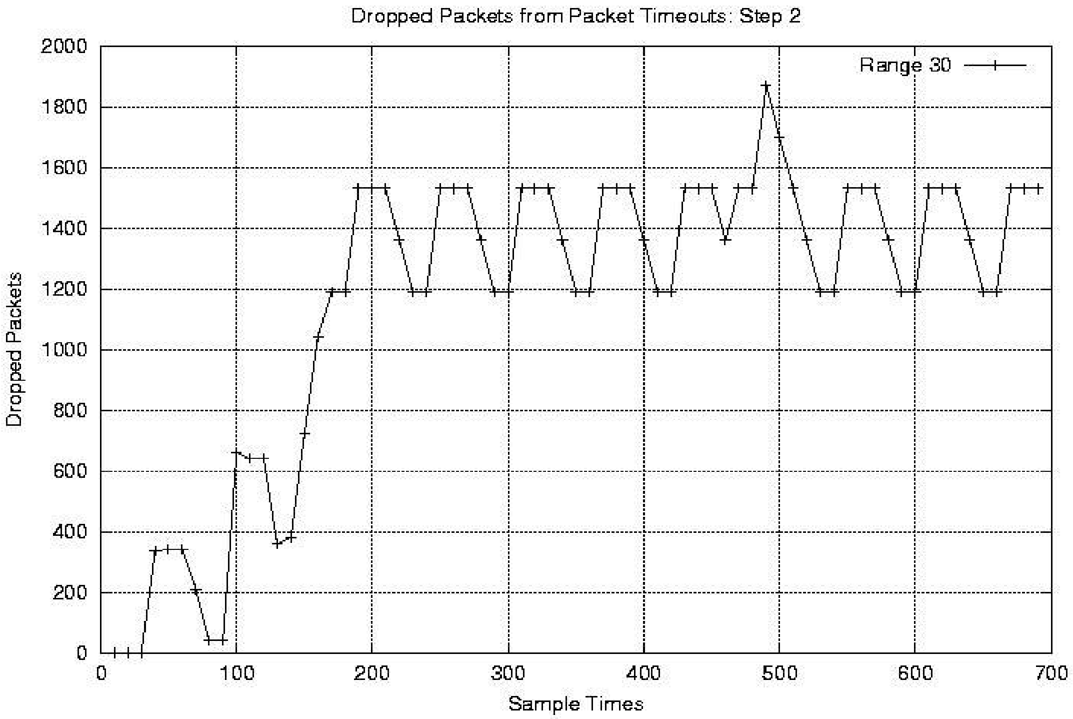


Figure 46: Step2 Packet timeout drop totals.

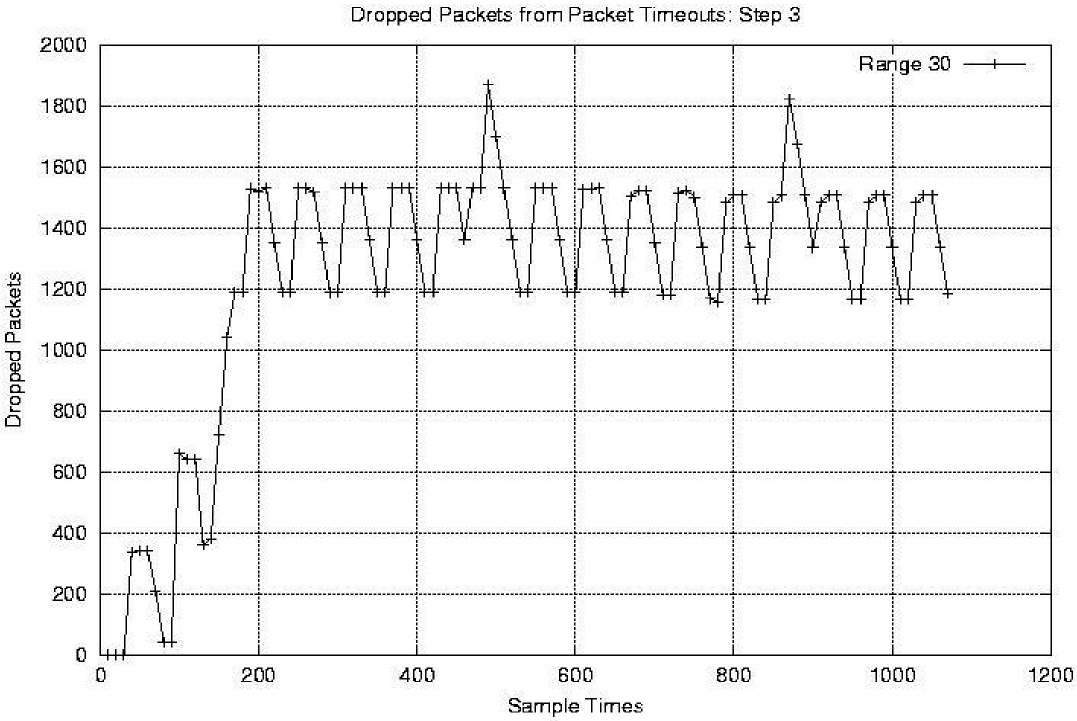


Figure 47: Step3 Packet timeout drop totals.

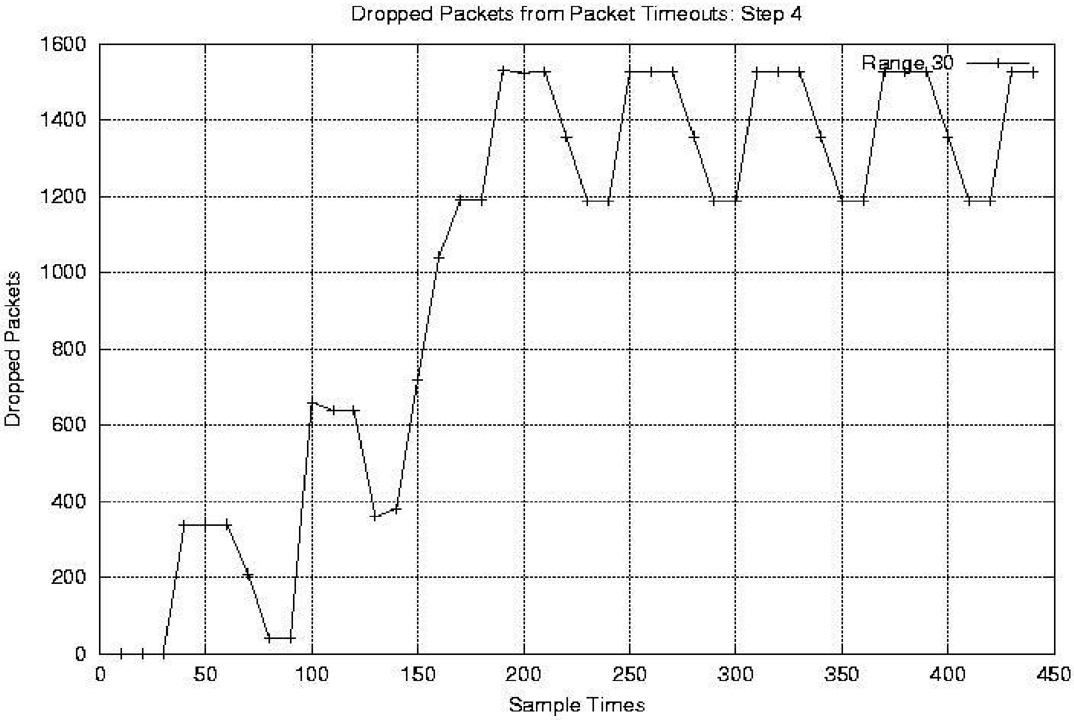


Figure 48: Step 4 Packet timeout drop totals.

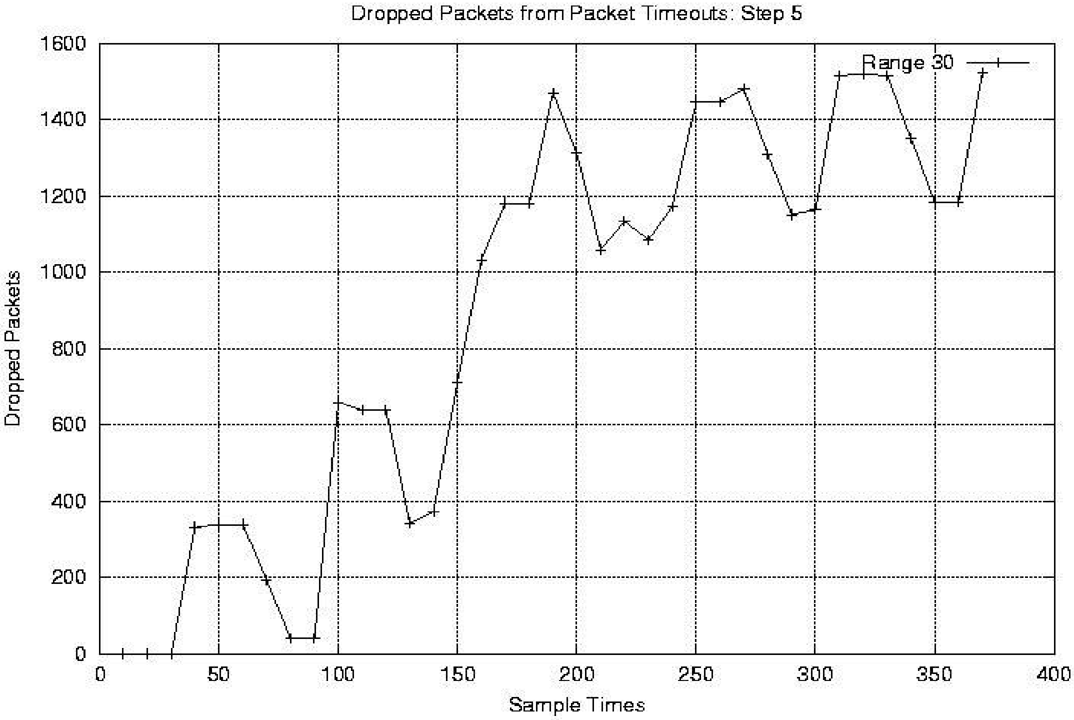


Figure 49: Step5 Packet timeout drop totals.

Figures 5-54 show the resulting throughputs for each simulation. We compared the throughputs for each transmission range at each step. Plots for Steps 1 and 2 do not show throughput for the 30 meter range because all of the data packets were dropped during the simulation.

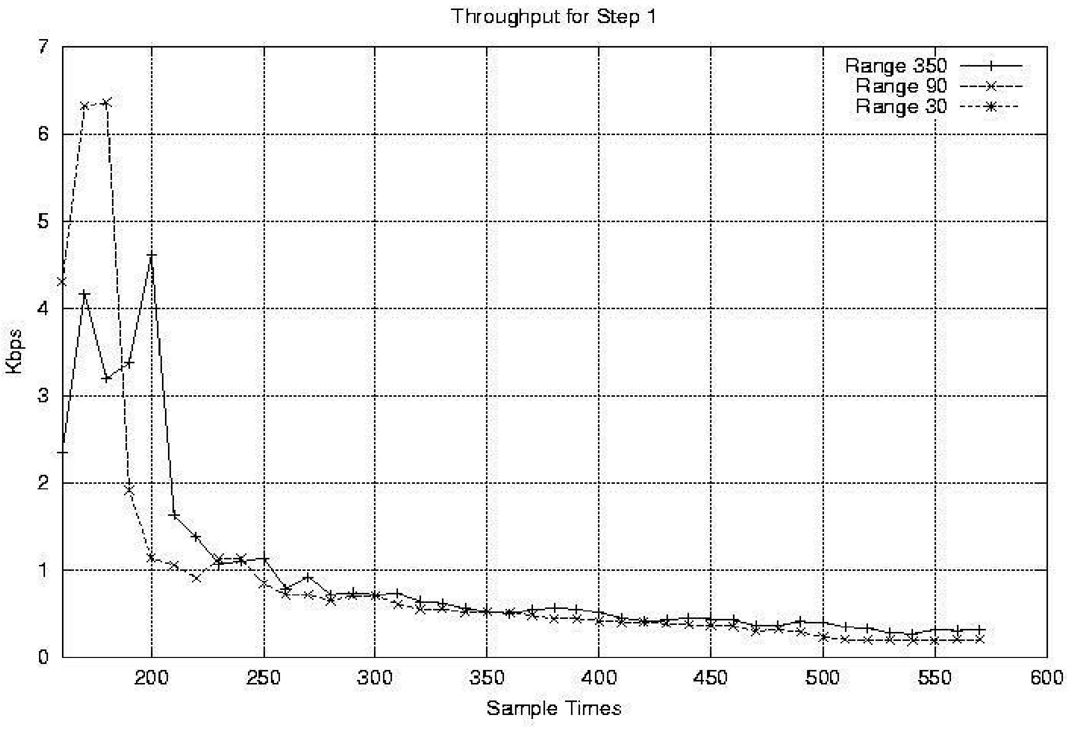


Figure 50: Step1 Throughput

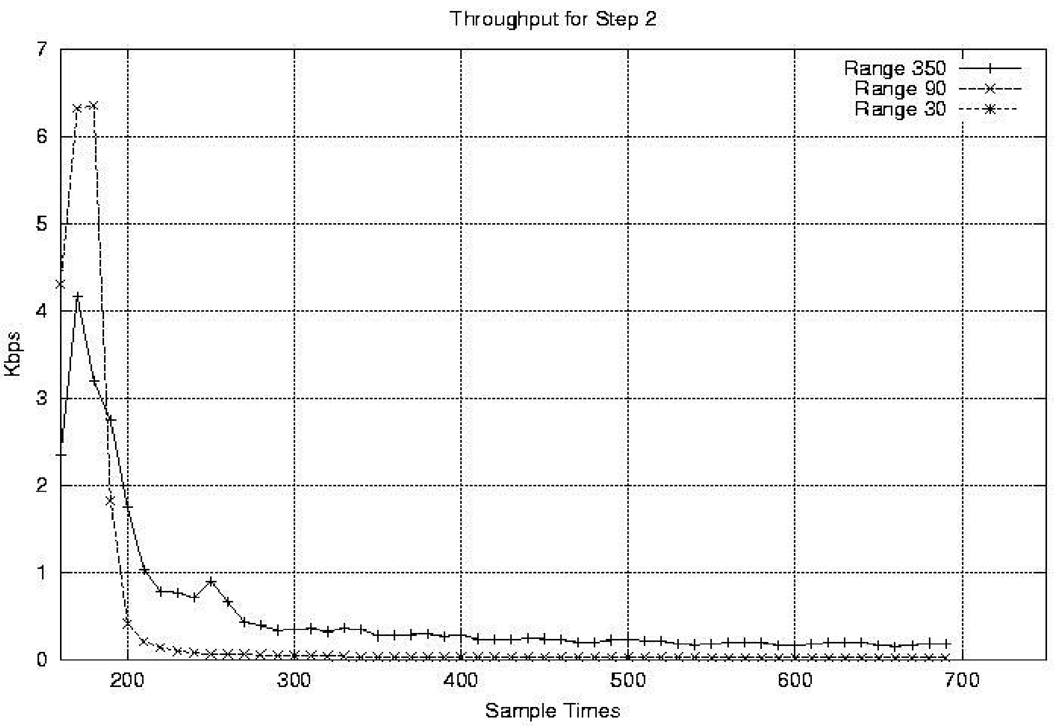


Figure 51: Step2 Throughput

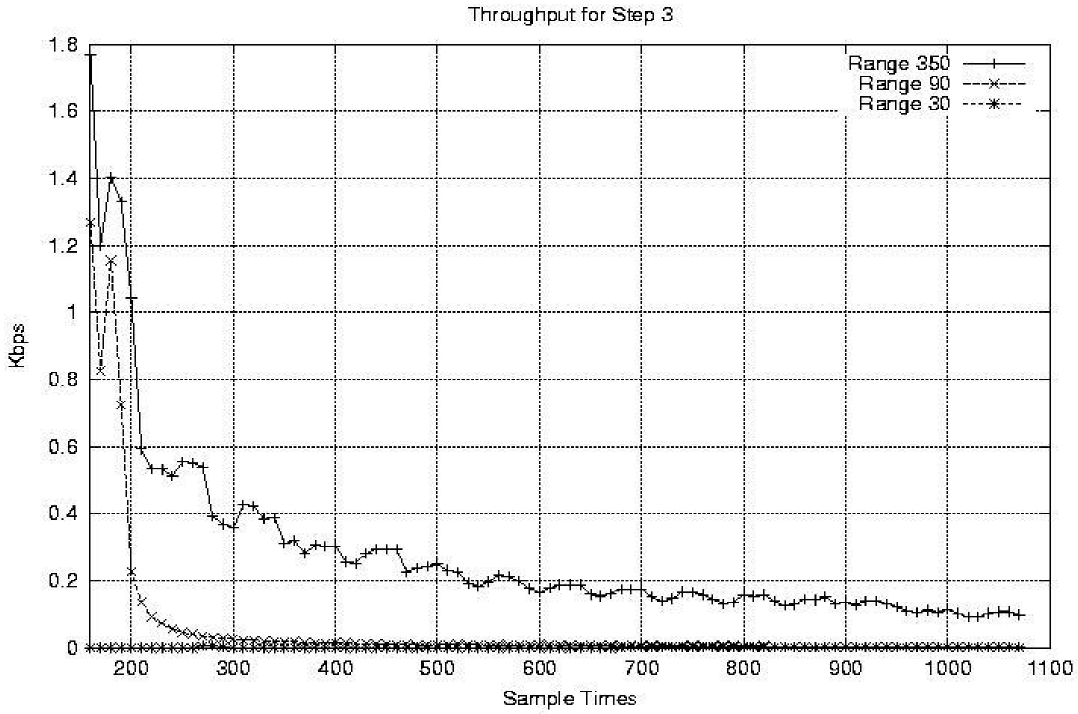


Figure 52: Step3 Throughput

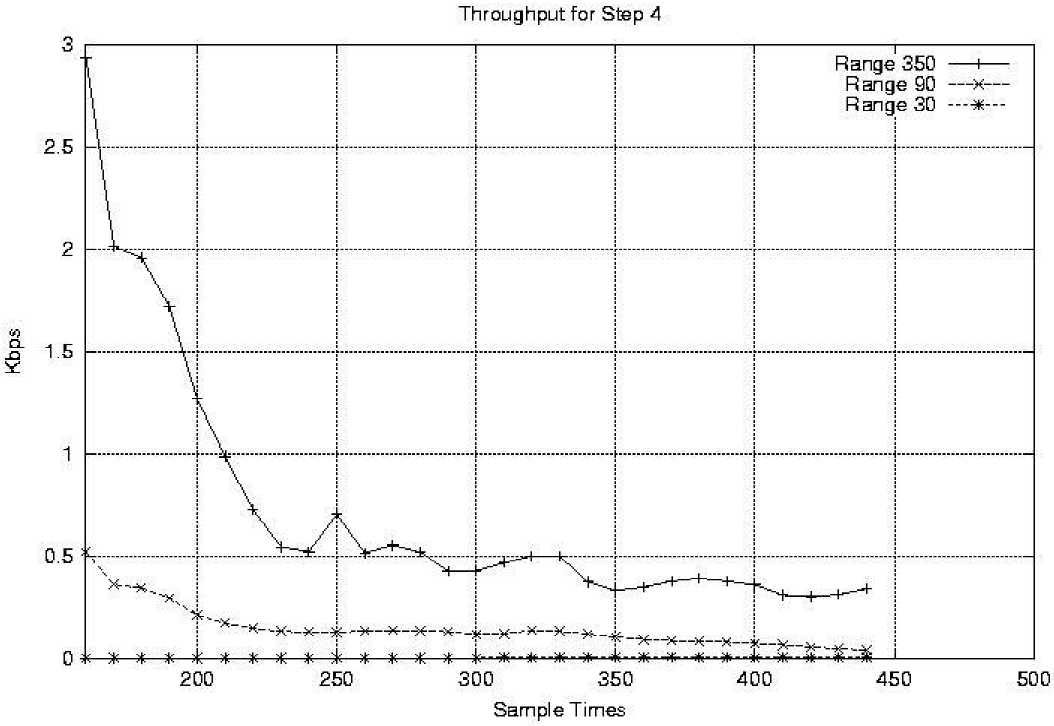


Figure 53: Step4 Throughput

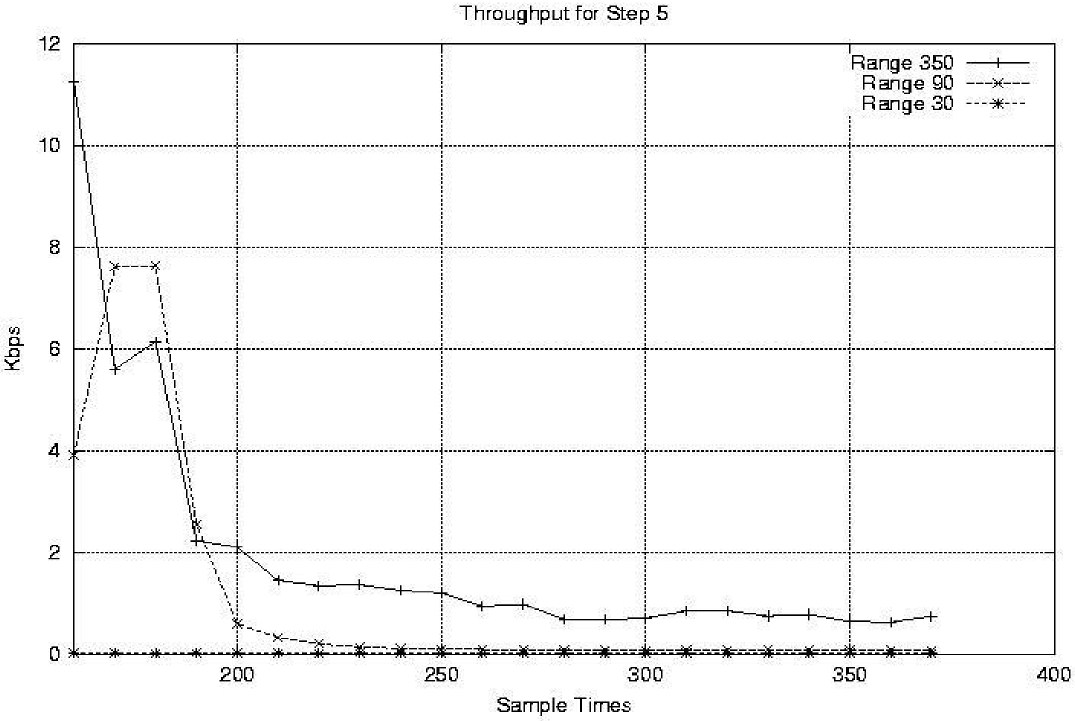


Figure 54: Step5 Throughput

6.4 Data Analysis

From our data, we have been able to show a link between having the MAC sense a metric, such as drops from collisions, and force a change to try to improve that metric and network quality. In this case, we show a reduction of the number of drops from MAC collisions as we reduce the transmission range. However, the throughput actually decreases when we decrease the transmission range, which we believe to be a result of an increase in the number of hops from source to destination. Additionally, reducing the transmission range too much can result in physically separate networks, as shown when the transmission range is 30 meters.

We are not sure why all of the packets were dropped in Steps 1 and 2 for a 30 meter transmission range. Individual soldiers are only about 10 meters apart, therefore, we should have seen at least a few packets received by immediate neighbor nodes. This could indicate a problem in the *ns-2* source code in the algorithm calculating the energy lost by a packet during transmission.

Despite our lack of success in finding a changeable attribute having a positive affect on network quality, we have been able to show a link between metrics which the MAC layer can sense and network quality, suggesting this is an area of further research and exploration. In addition, other metrics exist, such as packet drops from timeouts or a decrease in throughput as calculated from received packets, which the MAC layer can sense and use to make decisions. There are also other changes the MAC layer can force, such as changing from DCF to PCF mode or changing to a different frequency sub-band.

The primary contribution of this work was the analysis and modeling of Army "business practices" in the light infantry platoon. We were able to overlay on that a realistic network model usable for future exploration of wireless network protocols useful for military, civil defense, and emergency, first-responder applications. Using the light infantry platoon model, we will be able to develop any number of realistic scenarios to test our ideas about having the MAC layer act as an agent.

CHAPTER 7

SUMMARY AND CONCLUSIONS

7.1 Summary of Research Contributions

The purpose of this research was to create a modeling and simulation model and methodology, based on the use of the Unified Modeling Language for the conceptual analysis of the battlefield domain, and the use of the *ns-2* simulator for creating operational models for simulating the WLAN infrastructure on which model experiments can be executed. This work was undertaken as part of the larger ASOWN project, which has as its objective to make the 802.11b MAC layer wireless protocol more robust and flexible for use in a battlefield environment by focusing on use of adaptive, self-organizing agents for resource management at this layer of the network. By modeling the structure of a US Army light infantry platoon, their movement patterns, and message passing sequences, we were able to create a model for creating realistic scenarios for testing a WLAN. The platoon model helped us in designing source traffic applications and node movement files to define the scenario in *ns-2*. Using our defined software simulation model, we were able to conduct some intial tests about using the MAC layer as an agent, gathering local statistics and making local changes to affect the network quality as whole.

Our initial literature survey revealed many approaches to adaptive, self-regulating networks, to include mobile agents [MAE95][BRU91][GEN94] and adaptive routing

[JOD96][LSG00][LEE99]. Much of the mobile agent literature has some basis in adaptation based on ant colony optimization [HOL92][JOH01] where the agents moved along paths from node to node, gathering relevant statistics as they moved. However, none of the literature we found focused on agency at the MAC layer. Our work looks at how the node can be an agent itself and gather statistics from its local environment, both from events happening at the node and from information gleaned from incoming packets. In addition, we were not able to find any literature on a model of Army "business practices" at the light infantry platoon level. Our effort at modeling how Army procedures affect network traffic is essential in the development of communications platforms supporting military objectives. Finally, our work brought together many different, but related, fields in computer science, such as wireless networking and simulation, agents and adaptive systems, and conceptual modeling for novel domains. In none of the literature did we find a proposed solution crossing all of these domains.

7.2 Conclusions

In this thesis we have presented a discussion of the analysis activities, and creation of the domain model; transformed the domain model into a network simulation model; implementing the simulation model in *tcl* script for execution in the *ns-2* network simulation environment; evaluated the use of the model against battlefield scenarios; discussed the network metrics, data and patterns gleaned through experiments; and speculated on the use of this simulation model as an integral part of an exploratory methodology to examine a range of possible agent-based self-regulating behavioral enhancements to 802.11-based wireless networks.

7.3 Future Work

Future work can concentrate on one of three areas. The first area includes improving the infantry platoon model and adding some “front-end” onto the model for extensions, such as using Rational Rose® add-in capability to put a modeling front-end and tcl code generator to automatically generate the simulation model from conceptual model parameters. This will allow researchers to focus more on generating varied scenarios rather than writing the code to create the scenario in software.

The next area includes automating the interpretation of the trace files so as to extract appropriate pattern data over long simulation times (since the files can be on the order of hundreds of megabytes in size). This can reduce one of the problems we had of having the separate our scenario into separate steps. The result will be a better analysis of how the network reacts to events in the model after it has reached a steady state.

The final area for future work is to extend core 802.11 WLAN model, select a different *ns-2* variant model from another university, or find working examples of similar simulations to handle assigning frequency sub-bands, support for both DCF and PCF in a single simulation engine, a higher-level scripting interface to include more customizability of the protocol behaviors, and defining additional battlefield scenarios to be incorporated into the model to collect more network data. This last area is at the heart of the larger ASOWN research.

REFERENCES

[AXE97] Axelrod, R., *The Complexity of Cooperation: Agent-Based Models of Competition and Cooperation*, Princeton University Press, 1997.

[BIL02] Billsus, D., C. A. Brunk, C. Evans, B. Gladish and M. Pazzani, “Adaptive Interfaces for Ubiquitous Web Access”, *Communications of the ACM*, Vol. 45, No. 5, May 2002, pp. 34-38.

[BRE00] Breslau, L., D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y, Xu, and H. Yu, “Advances in Network Simulation”, *IEEE Computer*, Vol. 33, No. 5, May 2000, pp. 59-67.

[BRU02] Brusilovsky, P and M. T. Maybury, “From Adaptive Hypermedia to the Adaptive Web”, *Communications of the ACM*, Vol. 45, No. 5, May 2002, pp. 31-33.

[BRU91] Brustolini, J. C., “Autonomous Agents: Characterization and Requirements”, *Technical Report CMU-CS-91-204*, Carnegie Mellon University, 1991.

[EAB02] Bretz, E. A., “New Phones Play Games and Run Java Applications”, IEEE Spectrum, February 2002, pp. 62-63.

[DEN95] Dennett, D. C., *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. Simon and Schuster, New York, 1995.

[ETZ94] Etzioni and Weld, "A Softbot-Based Interface to the Internet", *Communications of the ACM*, Vol. 37, No. 7, July 1994, pp. 72-79.

[KFV00] Fall, K. and K. Varadhan, editors, *The ns Manual*, The VINT Project, www.isi.edu/nsnam/ns/ns-documentation.html, 24 August 2000.

[HDA01] *FM 7-8, Infantry Rifle Platoon and Squad*. Headquarters, Department of the Army, Washington, D.C. 22 Apr 1992 with Change 1, 1 Mar 2001.

[FRA96] Franklin, S. and A. Graesser, "Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents", Institute for Intelligent Systems, University of Memphis, 1996, HTML document, web site http://msci.memphis.edu/~franklin/AgentProg.html.

[GAS02] Gast, M. *802.11 Wireless Networks, The Definitive Guide,* O'Reilly & Associates, Inc., Sebastopol, CA. 2002.

[GEN94] Genesereth, M. and Ketchpel, "Software Agents", *Communications of the ACM*, Vol. 37, No. 7, July 1994, pp. 48-53.

[GOR99] Gordon, D., *Ants at Work: How an Insect Society is Organized,* The Free Press, 1999.

[GRM00] Greis, Marc. "Tutorial for the Network Simulator 'ns'." http://www.isi.edu/nsnam/ns/tutorial/index.html.

[HSM02] Hall, A., J. Surdu, F. Maymi, A. Deb, and K. Freberg. "Modeling the Communications Capabilities of the Infantry Soldier," *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[HAR00] Harvey, F., "The Internet in Your Hands", *Scientific American*, October 2000.

[HOL92] Holland, J., *Adaptation in Natural and Artificial Systems*, 2nd Ed., MIT Press, 1992.

[ISR02] Israelsohn, J., "Duking it on the Wireless Network", *EDN Magazine*, May 2, 2002, pp. 28-38.

[JOD96] Johnson, David B., and D. Maltz, "Dynamic Source Routing in Ad Hod Wireless Networks," *Mobile Computing*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[JOH01] Johnson, S., *Emergence: The Connected Lives of Ants, Brains, Cities and Software*, Scribner Publishing, Inc., 2001.

[JCS00] "Joint Vision 2020", Director for Strategic Plans and Policy, J5: Strategy Division, Joint Chiefs of Staff, U.S. Government Printing Office, Washington, D.C. June 2000.

[JRO01] "JROCM 134-01, Global Information Grid Capstone Requirements Document", Command, Control, Communications, and Computers, Plans, Policy and Projects Division, US Joint Forces Command, 30 August 2001.

[KUR03] Kurose, J. F., and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison Wesley Publishers, Inc., 2003.

[ANS99] LAN MAN Standards Committee of the IEEE Computer Society, ANSI/IEEE Standard 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[LSG00] Lee, S-J, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," ACM/Baltzer Mobile Networks and Applications, special issue on Multipoint Communication in Wireless Mobile Networks, 2000, to appear.

[LEE99] Lee, S. J., M. Gerla, and C. C. Chang, “On-Demand Multicast Routing Protocol”, in *Proceedings of IEEE WCNC’99*, New Orleans, LA, pp. 1298-1302.

[MAE94] Maes, P., “Agents that Reduce Work and Information Overload”, *Communications of the ACM*, Vol. 37, No. 7, July 1994, pp. 31-40.

[MAE95] Maes, P., “Artificial Life Meets Entertainment: Lifelike Autonomous Agents”, *Communications of the ACM*, Vol. 38, No. 11, November 1995, pp. 108-114.

[MSH02] Maymi, F., J. Surdi, A. Hall, R. Beltramini, “Modeling the Wireless Network Architecture of Land Warrior,” in *Proceedings of the 2002 Winter Simulations Conference.*

[NOP00] Noonan, R. and P. Cuviello, “C4 and ISR for the Objective Force,” www.us.army.mil.

[ORD01] “Operational Requirements Document for Land Warrior. ACAT I,” 31 Oct 2001. www.natick.army.mil/soldier/wsit/LW_ORD.pdf.

[PFL97] Pfleeger, C. P., *Security in Computing*, 2nd Ed., Prentice Hall Publishers, 1997.

[RUM86] Rumelhart, D. E. and J. L. McClelland, *Parallel Distributed Processing – Explorations in the Microstructure of Cognition: Fundamentals*, MIT Press, 1986.

[SAT01] Satyanarayanan, M., "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, August 2001, pp. 10-17.

[SHI99] Shinseki, E. "The Army Vision: Soldiers on Point for the Nation, Persuasive in Peace, Invincible in War," Office of the Chief of Staff, Washington, DC, October 1999 https://www.army.mil/vision/Documents/The%20Army%20Vision.PDF.

[ARM08] "TRADOC Program Integration Office Army Battle Command System: Requirements for Army C4ISR System", Technical Presentation, Combined Arms Center, Fort Leavenworth, KA. 2001. http://www.leavenworth.army.mil/tpioabcs/overview/slide1.htm.

[USA01]. "United States Army White Paper: Concepts for the Objective Force", Headquarters, Department of the Army, Washington, DC, U.S. Government Printing Office, November, 2001.

[WMT01] Woolcock, K., S. Murphy, P. Wyatt and M. Tyndall, "The Barbarians at the Gate: Wireless LAN Storms 3G Citadel", Nomura Equity Research, March 15, 2001, Nomura Securities Ltd., London.

<<add the additional references from Davis literature survey, part 2>>